

Rochester Institute of Technology

**RIT Scholar Works**

---

Theses

---

2-1-1996

## **Implementation of a real-time industrial web scanning system hardware architecture**

Lowell Ferguson

Follow this and additional works at: <https://scholarworks.rit.edu/theses>

---

### **Recommended Citation**

Ferguson, Lowell, "Implementation of a real-time industrial web scanning system hardware architecture" (1996). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact [ritscholarworks@rit.edu](mailto:ritscholarworks@rit.edu).

# **Implementation of a Real-Time Industrial Web Scanning System Hardware Architecture**

by

Lowell X Ferguson

A Project Submitted  
in  
Partial Fulfillment of the  
Requirements for the Degree of  
MASTER OF SCIENCE  
in  
Computer Engineering

Approved by:

---

Graduate Advisor James P. LeBlanc, Assistant Professor

---

Tony H. Chang, Professor

---

Roy S. Czernikowski, Professor and Department Head

Department of Computer Engineering  
College of Engineering  
Rochester Institute of Technology  
Rochester, New York  
February, 1996

# THESIS RELEASE PERMISSION FORM

## ROCHESTER INSTITUTE OF TECHNOLOGY COLLEGE OF ENGINEERING

Title: Implementation of a Real-Time Industrial Web Scanning System Hardware Architecture

I, Lowell X Ferguson, hereby grant permission to the Wallace Memorial Library to reproduce my project in whole or part.

Signature: \_\_\_\_\_

Date: 2/26/96

# ABSTRACT

Industry manufactures many products in a continuous web process, such as paper, plastics, and metals, all of which must be inspected. One means of inspecting them is with imaging sensors and electronic circuitry configured as a web scanning system.

The hardware architecture of a web scanning system is discussed. An analysis of the system needs shows a requirement for the system to be both real-time and capable of scanning in an industrial environment. An implementation of one such system that meets these qualifications, and is designed for use with higher end web scanning applications, is reviewed. Architectural changes, or additions, due to continued technology advancement for the existing system, which meet the previously identified system requirements, are identified.

Definition of the problem statement includes a description of a continuous web manufacturing process. The demonstrated need for inspection of webs is discussed, along with human inspection techniques employing statistical sampling and segment testing. Technology that may be employed for the automated inspection of web products is described.

The architectural description is that of the Veredus Quality Control System. Potential improvements to the Veredus System are described, which would improve upon the continuous flow-through design architecture. Conclusions address the successfulness of the Veredus system, and the feasibility of the proposed improvements.

The following names used here and in the remainder of the document are registered trademarks of the respective companies:

TAXI

AMD Corporation

© by Lowell X Ferguson, 1996

All rights reserved

# Table of Contents

THESIS RELEASE PERMISSION FORM	ii
ABSTRACT	iii
TABLE OF CONTENTS	v
LIST OF FIGURES	vii
LIST OF TABLES	viii
LIST OF EQUATIONS	ix
GLOSSARY AND ACRONYMS	x
<b>1. THEORY</b>	<b>1-1</b>
1.1. Description of a Web Process	1-1
1.2. Human Inspection Techniques	1-5
1.3. Inspection Requirements	1-7
1.4. Imaging Sensor Technologies	1-11
1.5. Frame-Based Versus Line-Based Scanning	1-15
1.6. Production Line Data Acquisition	1-20
1.7. Image Processing	1-24
1.8. Imperfection Extraction	1-36
1.9. Verification Aids	1-42
<b>2. IMPLEMENTATION</b>	<b>2-1</b>
2.1. Architectural Overview	2-2
2.2. Interboard Connections	2-5
2.3. Board Format	2-9
2.4. Board Functions	2-11

<b>2.5. Data Acquisition</b>	<b>2-19</b>
<b>2.6. Camera to System Data Interface</b>	<b>2-22</b>
<b>2.7. Data Correction</b>	<b>2-25</b>
<b>2.8. Edge Tracking</b>	<b>2-28</b>
<b>2.9. Enhancement</b>	<b>2-33</b>
<b>2.10. Thresholding</b>	<b>2-38</b>
<b>2.11. Data Reduction</b>	<b>2-44</b>
<b>2.12. Connected Components and Classification</b>	<b>2-46</b>
<b>2.13. Interprocessor Communication</b>	<b>2-49</b>
<b>2.14. Event Tracking</b>	<b>2-51</b>
<b>2.15. Diagnostics</b>	<b>2-52</b>
<b>2.16. Gray Scale Display</b>	<b>2-57</b>
<b>3. IMPROVEMENTS</b>	<b>3-1</b>
<b>3.1. Logic Compaction</b>	<b>3-1</b>
<b>3.2. Backplane Communications</b>	<b>3-4</b>
<b>3.3. Connectivity and Measurement</b>	<b>3-9</b>
<b>3.4. Classification</b>	<b>3-13</b>
<b>3.5. Diagnostics</b>	<b>3-15</b>
<b>3.6. Display Subsystem</b>	<b>3-17</b>
<b>3.7. Triple Banked Memories</b>	<b>3-19</b>
<b>4. CONCLUSION</b>	<b>4-1</b>
<b>5. LITERATURE CITED</b>	<b>5-1</b>

# List of Figures

FIGURE 1-1: EXAMPLE WEB PROCESS.....	1-1
FIGURE 1-2: A WEB PRODUCT ACCUMULATOR .....	1-3
FIGURE 1-3: A FLYING SPOT LASER .....	1-12
FIGURE 1-4: AN OPERATIONAL SUMMARY OF A CCD SENSOR .....	1-13
FIGURE 1-5: CCD SENSOR CONFIGURATIONS .....	1-14
FIGURE 1-6: FRAME OVERLAP IN FRAME GRABBERS .....	1-17
FIGURE 1-7: TRANSMISSION AND REFLECTION MODES .....	1-18
FIGURE 1-8: A CONTINUAL FLOW-THROUGH PROCESS.....	1-19
FIGURE 1-9: A GENERAL WEB SCANNING ARCHITECTURE .....	1-25
FIGURE 1-10: ORIENTATION OF CONVOLUTION VARIABLES .....	1-26
FIGURE 1-11: A GENERALIZED FIR FILTER IMPLEMENTATION .....	1-27
FIGURE 1-12: A GENERALIZED IIR FILTER IMPLEMENTATION .....	1-28
FIGURE 1-13: A GENERIC PIPELINED LOOK-UP TABLE .....	1-33
FIGURE 1-14: EXAMPLE FEATURE MEASUREMENTS .....	1-39
FIGURE 2-1: BACKPLANE DATA FLOW CONFIGURATION .....	2-2
FIGURE 2-2: CONTINUAL FLOW-THROUGH PROCESS.....	2-3
FIGURE 2-3: INTERCONNECTION OF HOST RACK .....	2-4
FIGURE 2-4: HIGH-SPEED SLAVE/MASTER CONFIGURATIONS .....	2-7
FIGURE 2-5: VEREDUS BOARD FORM FACTORS .....	2-10
FIGURE 2-6: VEREDUS PIPELINE BLOCK DIAGRAM .....	2-16
FIGURE 2-7: EXAMPLE VEREDUS PIPELINE CONFIGURATIONS.....	2-17
FIGURE 2-8: A VEREDUS 1000 CONFIGURATION WITH OLDER GENERATION BOARDS .....	2-17
FIGURE 2-9: A VEREDUS 250 CONFIGURATION WITH NEWER GENERATION BOARDS .....	2-18
FIGURE 2-10: DIGITAL CAMERA TRANSMITTER CONNECTIONS.....	2-20
FIGURE 2-11: CAMERA SUBSYSTEM WITH EXPOSURE CONTROL .....	2-21
FIGURE 2-12: CAMERA TO SYSTEM MODULARITY .....	2-23
FIGURE 2-13: DIGITAL CAMERA MULTIPLEXOR DAUGHTER BOARD CONNECTIONS.....	2-24
FIGURE 2-14: DOUBLE BANKED RAM GAIN TABLE .....	2-27
FIGURE 2-15: MULTIPLE GAIN SECTIONS CONNECTIONS.....	2-28
FIGURE 2-16: A/D AND IFB EDGE TRACKING ZONES .....	2-30
FIGURE 2-17: IMPLEMENTED EDGE TRACKING DATA FLOW .....	2-31
FIGURE 2-18: IMPLEMENTATION OF 4-PIXEL WIDE ONE-DIMENSIONAL PIPELINED CONVOLUTION .....	2-34
FIGURE 2-19: IMPLEMENTATION OF 4-LINE TWO-DIMENSIONAL PIPELINED CONVOLUTION .....	2-34
FIGURE 2-20: CONVOLUTION, BARREL SHIFTER, AND CLIPPER DATA FLOW.....	2-35
FIGURE 2-21: PARALLEL CONVOLUTION IMPLEMENTATIONS .....	2-37
FIGURE 2-22: DOUBLE BANKED THRESHOLD TABLE .....	2-39
FIGURE 2-23: TILING FUNCTION IMPLEMENTATION.....	2-42
FIGURE 2-24: RLE-I ENCODING FORMAT .....	2-44
FIGURE 2-25: RLE-II ENCODING FORMAT .....	2-45
FIGURE 2-26: DIAGNOSTIC INPUT, DIAGNOSTIC OUTPUT, AND BYPASS IMPLEMENTATIONS .....	2-54
FIGURE 2-27: BIST CONCEPTUAL IMPLEMENTATION .....	2-56
FIGURE 2-28: EXAMPLE MODES OF SAMPLING DISPLAY DATA FROM A SCAN LINE .....	2-59
FIGURE 2-29: CAPTURE OF ENHANCED AND UNENHANCED DATA .....	2-60
FIGURE 2-30: FEATURE FINDER IMPLEMENTATION .....	2-61
FIGURE 3-1: CONCEPTUAL POINT-TO-POINT INTERCONNECTION.....	3-5
FIGURE 3-2: SAMPLE POINT-TO-POINT INTERCONNECTION BOARD.....	3-7



# List of Tables

TABLE 1-1: EXAMPLES OF PRODUCT, WIDTH, AND SPEED .....1-1

# List of Equations

*EQUATION 1-1* .....1-25  
*EQUATION 1-2* .....1-27  
*EQUATION 1-3* .....1-29  
*EQUATION 1-4* .....1-35  
*EQUATION 1-5* .....1-35  
*EQUATION 1-6* .....1-36

# Glossary and Acronyms

A/D	Analog-to-Digital Conversion Board (p. 2-11)
ACB	Analog Control Bus (p. 2-8)
ASIC	Application Specific Integrated Circuit (p. 3-2)
BIST	Built In Self Test (p. 1-10)
C&M	Connectivity and Measurement Board (p. 2-14)
CCD	Charged Coupled Device (p. 1-12)
CDS	Correlated Double Sampling (p. 1-21)
Constant Frequency	A CCD and system scanning mode in which the line transfer clock is always driven at the same frequency. As the production line changes speed, the area scanned by each CCD output line changes (pp. 1-21, 2-20)
Constant Pixel	A CCD and system scanning mode in which the down-web pixel size of all pixels is maintained at the same value. It requires the line transfer clock to be derived from a rotary encoder signal (pp. 1-21, 2-20)
Conv 1.0	Convolver board version 1.0, or Convolver 1.0 (p. 2-13)
Conv 3.0	Convolver board version 3.0, or Convolver 3.0 (p. 2-13)
Correlated Double Sampling	A process used in establishing the value to digitize in a CCD array camera, where two samples are taken, and the difference between the two is the appropriate signal (p. 1-21)
DCM	Digital Camera Multiplexor Board (p. 2-12)

DCR	Digital Camera Receiver Board (p. 2-11)
DCT	Digital Camera Transmitter Board (p. 2-11)
DI/O	Host level Digital Input/Output Board (p. 2-16)
DSP	Digital Signal Processing
E-net	Host level Ethernet adapter board (p. 2-16)
Feature Vector	The resultant output of Connectivity and Measurement process on the Veredus Quality Control System (p. 2-3, 2-47)
FIFO	First In First Out memory
FIFO Board	A Board which provides a FIFO buffer (p. 2-13)
FIR	Finite Impulse Response filter (p. 1-28)
FPGA	Field Programmable Gate Array (p. 3-3)
GRAPH	Host level Color graphics interface board (p. 2-16)
GRAY	Gray Scale Display Board, Version 1.0 (p. 2-15)
GRAY 3.0	Gray Scale Display Board, Version 3.0 (p. 2-15)
Host	Commercially available circuit boards providing the system wide controlling microprocessor functions. These include disk and tape drive interfaces, color graphic interfaces, ethernet interfaces, etc. (p. 2-4)
Host Microprocessor	The microprocessor which dictates what all other portions of the system will do. It is responsible for accurate configuration of all pipeline boards, and receives all imperfection data from reporting data sources. (p. 2-4)
HSM	High-Speed Master (p. 2-6)

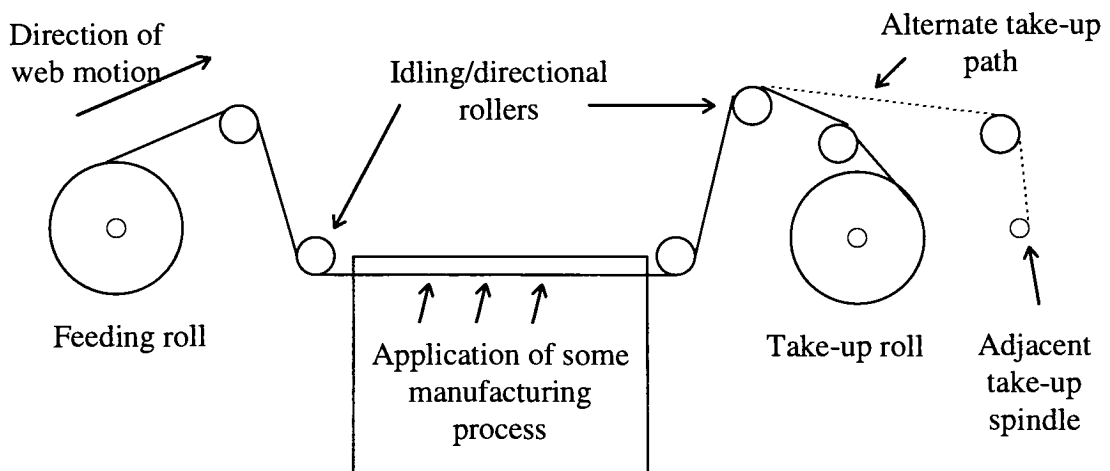
HSS	High-Speed Slave (p. 2-6)
I/F	Camera interface board (p. 2-11)
IC	Gray Scale 3.0 Image Channel Board (p. 2-15)
IFB	Intelligent Filter Board (p. 2-12)
IIR	Infinite Impulse Response filter (p. 1-28)
LFSR	Linear Feedback Shift Register (p. 2-55)
LUT	Look-Up Table (p. 1-32)
MHz	Megahertz
NRE	Non-Recurring Engineering charges (p. 3-3)
ns	nanosecond
Phase	Thresholded value indicating if a pixel is in tolerance or out of tolerance. If out of tolerance, a level is included to show how far out of tolerance it is (p. 1-33)
Quad II	A circuit board with four parallel Motorola 68010 microprocessors on it (p. 2-14)
Quad III	A circuit board with four parallel Motorola 68020 microprocessors on it (p. 2-14)
RLE 1.0	Run Length Encoder board version 1.0, or Run Length Encoder 1.0 (p. 2-13)
RLE 1.6	Run Length Encoder board version 1.6, or Run Length Encoder 1.6 (p. 2-13)
RLE-I	A Run Length Encoding scheme that counts sequential pixels at the same phase (p. 2-44)

RLE-II	A Run Length Encoding scheme that reports phase changes in absolute cross-web and down-web values (p. 2-45)
RPTR	Host Bus repeater board set (p. 2-16)
SA	Signature Analyzer (p. 2-55)
TDI	Time Delay Integration (p. 1-13)
WTRK	Web Track Encoder Board (p. 2-15)

# 1. Theory

## 1.1. Description of a Web Process

Many materials are manufactured using a continuous web process. As shown in Figure 1-1, a web process consists of the manufacturing of a flat product in a continuous ribbon of material, typically in a wide width format and at high speed. For examples of product, width, and speed, see Table 1-1.



**Figure 1-1: Example Web Process**

Material	Product Width	Web Speed
Paper	200 to 300 inches	1000 feet per minute
Tin Plate Steel	55 inches	1000 feet per minute
Plastics	50 inches	200 feet per minute
Aluminum	70 inches	3000 feet per minute

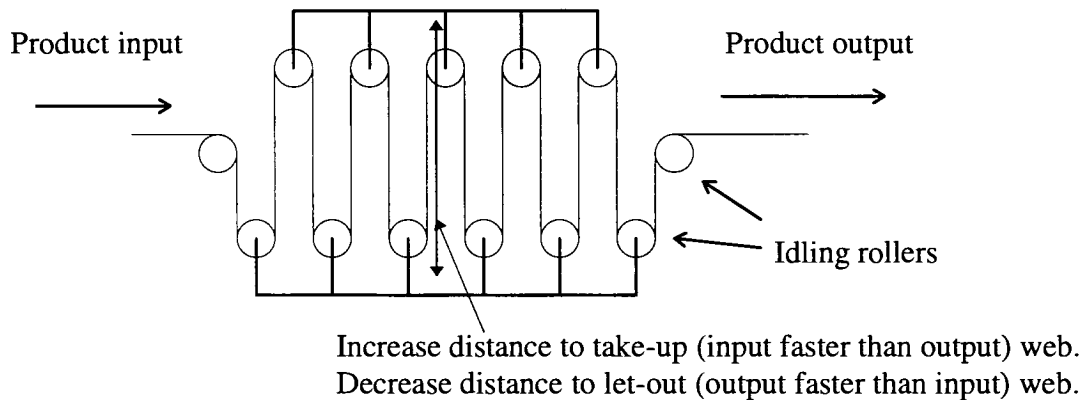
**Table 1-1: Examples of Product, Width, and Speed**

As these web materials are manufactured, material is moved from the feeding roll to the take-up roll. The feeding roll, however, can take on multiple forms. In a paper manufacturing process, the feeding roll can be a conveyor, on which paper pulp is deposited. As the conveyor moves forward, water is removed from the pulp in several manners, ultimately resulting in the squeezing of the paper between two rollers to obtain uniform thickness and width. The resultant continuous stream of paper arrives on the take-up roll, in lengths of 50,000 feet or more. The production line may momentarily slow to allow a mature roll on the take-up reel to be removed, while a new roll is started on an adjacent take-up spindle. The ultimate result of this process is a production line with paper continuously moving on it.

Tin-plated steel is manufactured in more of a continuous speed manufacturing process. The raw rolled steel is subjected to an acid bath to remove any contaminants prior to passing through an electroplating bath of tin. A fairly constant material speed is required to avoid overexposing the rolled steel to either the acid bath, or the tin plate. This is resolved by placing an accumulator between the main portion of the production line, and the feeding or take-up roll. As shown in Figure 1-2, material can be fed into the accumulator faster than it is removed by increasing the distance between two series of rollers. As the distance increases, the speed of the material fed in must equal the rate material flows out, plus the distance increase multiplied by the number of rollers in the accumulator. In the feeding roll case, this allows the feeding end of the process to build a buffer of material to feed from, while a new feeding roll is installed ahead of the accumulator. Once no material is entering the feeding accumulator, the distance between



the accumulator rollers is decreased as needed to allow material to flow through the main portion of the production line, at only a slightly (10-20%) reduced rate. By the time the buffer in the accumulator is consumed, a new roll is in place and feeding the accumulator, allowing the production line to run continuously. A similar situation occurs, in reverse, at the take-up end.



**Figure 1-2: A Web Product Accumulator**

The existence of an accumulator on a web manufacturing line is significant to the inspection of the product. An on-line inspection is most desirable as late in the manufacturing line as possible, whether the inspection is by human inspectors, or by machine vision products. Hence, if an accumulator exists on the production line, the place inspection will likely be desired is after the accumulator. This results in the inspection area having speeds that can fluctuate massively, from nearly no movement, to near twice the normal operating speed of the general production line. The inspection process is complicated by this requirement, requiring a machine vision system to handle web speed fluctuations, and higher web speeds than the general production line.

Various imperfections can be found in manufactured webs. Some of these are consistent across all continuous web processes, such as repeating imperfections generated by a roller with a blemish on it. The negative of the blemish can be pressed onto the web in progress once on every revolution of the roller in question. Determination of the distance from one instance of this repeating imperfection to the next, allows determination of the roller diameter, and hence, reduces the number of locations to search for the roller blemish. A binding roller is also a common ailment to web processes, effectively generating a scrape mark on the manufactured web. Once again, characteristics of the scrape can help establish where the failing roller is located.

Other imperfections are unique to particular products. Paper suffers from holes, dirt, and fibers in the web. Tin-plate steel may have inconsistent plating because of malfunctions in the electroplate area. Establishing the characteristics of particular imperfections allows for faster correction of the process, resulting in shorter down-times for the production line, and lower amounts of product waste.

## **1.2. Human Inspection Techniques**

Web processes have relied on human inspectors since inception. Tools may be employed to help measure the size or spacing of imperfections, but the ultimate image processing engine is the human mind. Effectively this results in the use of either statistical quality control (potentially through the use of end testing), or on-line inspectors gazing at product for hours on end.

Statistical quality control requires the sampling of manufactured product, typically removing some small portion (less than 1%) from the end of a completed roll of material, and inspecting it. This typical case is referred to as end testing. It may be necessary to perform some process on the material to help bring out any existing imperfections. For instance, film may have to be developed to establish the imperfections. Paper can be treated with a special coating to allow imperfections to stand out. No matter what the case, human inspectors must then scour the sampled product for imperfections. Obviously this requires time, both from the aspect of labor involved with the inspection process, and time before a manufactured product can be certified as inspected. The first runs the risk of inconsistency caused by fatigue, and subjective variations between inspectors in determining what is a good product, and what is not. The second results in the need for a standing inventory while waiting for inspection results, and adds to the possibility of additional imperfect product being manufactured during the inspection time period [1].

Additionally, the end testing is sampling less than 1% of the produced product. It is possible imperfections are generated only within the uninspected 99%, and not are detected until used by the end user. Detection at this level is viewed as too late by some manufacturers [1]. Note that the inspected “less than 1%” is destructively tested as well, implying the shipped 99% of product is not inspected. Transition to quality concepts in many manufacturing plants (requiring inspection of shipped product), is rapidly becoming the normal mode of operation. Steel manufacturers have been paid premiums for product that has been 100% inspected. Paper manufacturers producing new products have been allowed a market by their buyer only under the condition the product be 100% inspected.

Perhaps the best example comes from reiterating the manufacturing process of a web product near the end of the roll. The process slows (even somewhat with an accumulator) at the end of the rolls. Some production line malfunctions only occur when the web is moving at normal operating speed, such as scraping of the product because of a roller which sticks at high speed, but freely rolls at lower speeds. Statistical sampling, because it typically uses end tests, would sample the portions of the web where the roller works properly.

As a result, on-line inspection is frequently required. Slower speed web processes can complete this function with inspectors simply staring at the manufactured web for one to two hour shifts. Faster webs (i.e., tin-plate steel) have employed strobe lights flashing at the web as it moves past the inspection station. Providing the equivalent of a momentary

snapshot of the web at each flash of the strobe, an inspector can, with some level of success, inspect a web moving at over 1,000 feet per minute. At most, two inspectors can work on a single side of a 55 inch wide web, because of the physical limitations of the width of a human versus the web. This results in an inspector observing over 5,400 square inches of material per second at 1,000 feet per minute. Imperfections that occur regularly over time can be detected in this manner. This clearly will not assure detection of the random occurrence imperfection, and does not guarantee detection of all repeating imperfections. This process still suffers from inspector fatigue and variation between inspectors identified above.

### **1.3. Inspection Requirements**

Generally, inspection of a web must include not only a statement of good or bad, but also an indication of why it is good or bad. If an imperfection is detected, the goal is to identify the source of the imperfection, and correct the process as quickly as possible. To classify the detected imperfection, as to its origin, requires inputs such as deviation to the high side of the background level (a bright spot) or low side (a dark spot). How wide and long is the imperfection? Does it repeat? If it does repeat, what is the separation (pitch) of the imperfection stream? Are there multiple small fluctuations, which when combined together form one general imperfection, instead of a bunch of small ones? What does it physically look like, i.e., a spot, or a dent? These characteristics are employed by human inspectors when determining the nature of an imperfection. It must be expected that any automated web scanning system can do the same. In order to accomplish this, enhancements to the data are frequently required to pick out fine details, i.e., automatic

internal adjustments caused by light level variations, edge enhancement, etc. These results are then subjected to some form of “connecting the dots” to establish the identity of the detected imperfection. Further analysis may then be needed to establish the repeating nature of the imperfection, or the locality of other imperfections.

A further complication for an automated web scanning system are the hard “deterministic” requirements of what must be detectable. With human operators, the size of a detected imperfection is somewhat subjective. It is difficult to know if a detected imperfection is five millimeters (mm), or six. Detection and classification of the imperfection is all that matters. With an automated scanning system, a hard requirement must be established for the minimal imperfection, which must be flagged, because the subjective human mind will not be the analysis engine used on the scanning system’s detector. The result is web process manufacturers’ requirement for the detection of imperfections 1 mm in size on webs 8-10 feet wide, and running at up to 6,000 feet per minute[6]. Dependent upon the fidelity of the employed sensor systems, and observed signal-to-noise ratios, the scanning system requires an image granularity of around 1 mm, down to a nyquist principle requirement of 0.5 mm (or about 0.020 inches).

It can be argued that the required detectable imperfection size from web process manufacturers is finer than needed. This can only be proven in a case-by-case study of a particular product to be scanned. As a result, a scanning system must be capable of more than just meeting the high resolutions (small image granularity) initially required of web process customers. It must also be modular enough to reduce the resolution (increase

granularity) to meet goals adjusted by experimentation. Presumably, the reduction in resolution also affords a significant reduction in price of the scanning system.

The result of these varied scanning system requirements is to pursue a digital electronics solution for the main scanning engine. Image enhancements are easier to implement consistently and with minor adjustment capabilities in the digital realm. This is especially true if multiple sensors or analysis functions provide information, which is pulled together to identify a particular imperfection. Providing measurements in the digital realm is fairly straightforward, giving cross-web (width of the web) and down-web (length) measurements from some arbitrarily set datum point easily. Databasing of detected imperfections can occur readily, including complete cross-references to the time of day, roll number, target customer, and chief line operator.

Once a scanning system is installed, training and certification must occur. Basically, this requires assuring that the scanning system observations are consistent with the human inspectors. During training, the scanning system will usually identify an imperfection's location. That area is then viewed, verified to have an imperfection, and then the imperfection is classified by human operators. This input is then fed back into the scanning system. Once enough training has occurred so that the scanning system is receiving little additional feed-back for adjustments, certification will occur. Classified outputs from the scanning system are then compared to those of the human operators, and if the two match for a sufficiently long period of time, the system is termed certified.

Unfortunately, the act of training can be painful. One of the strong arguments against human inspectors is the inconsistency of imperfection classification between inspectors. Now in training the scanning system, these same inspectors are feeding information to the scanning system, which may be contradictory. Frequently, a panel of inspectors must judge the imperfection, and an agreed to classification reached prior to entry into the scanning system [6]. Clearly this shows that the ability of a scanning system to properly classify detected imperfections will be no better than the ability of the human inspectors who have trained it. However, once certified, it should not suffer from the fatigue or inconsistency of human operators.

The scanning system needs some sort of self-diagnostic to assure proper responses of the scanning system for a known input stimulus. This takes on two forms: the first requires the provisions for diagnosing digital circuit boards. Through the use of Built-In Self Test (BIST), and functional testing through diagnostic data ports, conceptual means exist to provide these functions readily. Establishing a way to access the diagnostic features must then be provided by software in a user friendly way. This sounds simple, but yet is implemented altogether too infrequently because of delivery schedule problems, lack of real estate on circuit boards, or failure to understand the need for such features.

The second, and larger, issue at hand is a means to feed a known stimulus into the imaging sensor, or plane, and obtain a known output. Conceptually, provide a fake imperfection where the imaging sensor is focused with characteristics similar to those of an imperfection, and assure the system detects it. One of the primary requirements for

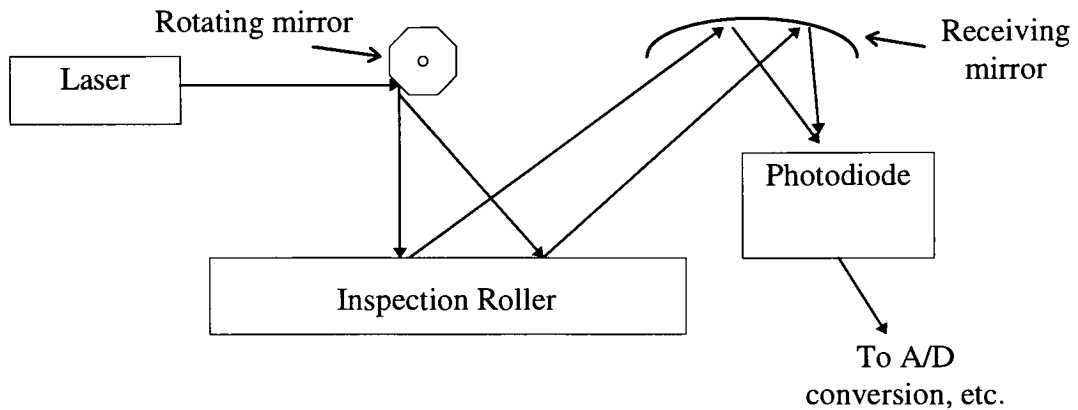


this input stimulus device is that it cannot interfere with the normal scanning operation of the system when it is not in the diagnostic mode. Part of the problem is that physical space on the production line is always at a premium, especially near the area where the sensor is focused. This makes it difficult to generate a known input, which will maintain the focal length of the imaging sensor, and still not interfere with normal scanning operations. Frequently, the stimulus takes the shape of a test target that is wrapped on the inspection roller, and rotated by hand to reach the sensor inspection point. While functional for the initial set-up of a system, it does not provide a routine diagnostic capability, which production line quality assurance personnel can execute once a day to assure the system is consistent.

#### **1.4. Imaging Sensor Technologies**

Several methods of acquiring sensory input from a web product exist. Flying spot lasers project a laser onto a precision ground mirror assembly, typically with eight mirrors on it, as shown in Figure 1-3. This assembly is rotated with high precision, allowing the reflected laser beam to project onto the web product. The signal resulting from the laser striking the web is received through a mirror assembly into a photosensitive device with the same detection wavelength the laser is generating. Electronics synchronize the signal received from the laser to the angle of the rotating mirror to determine which part of the web the “flying spot” was illuminating. This output can then be digitized and processed as needed. This technology provides high-fidelity signals, and works well in the near infrared range (excellent for sensitized products). But it requires a very elaborate set-up

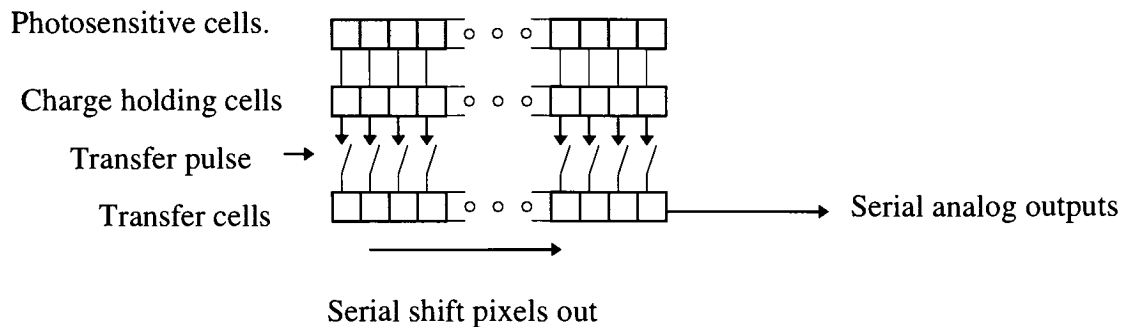
on the production line, including a shroud over the scanning apparatus to keep out stray light, which makes it difficult to install in already existing production lines.



**Figure 1-3: A Flying Spot Laser**

An alternative sensing device is a charged-coupled device, or CCD, camera. These devices generate an electronic signal proportional to incident light in the range of 200 to 1100 nanometers. A series of silicon elements are located next to each other, as shown in Figure 1-4, which receive light from some focusing means. As light strikes the elements, a charge proportional to the intensity of the light accumulates in a charge holding cell located near the element. With the receipt of a periodic start or transfer pulse, the charge located in the holding cells is transferred from each element into a transfer cell. At the completion of the transfer cycle, the holding cell contains no charge. The cell then starts to integrate again while waiting for the next transfer pulse. Meanwhile, the charges transferred into the transfer cells are serially read out in a shift register pattern at a rate determined by clock pulses provided to the image array. The output is a discrete time analog representation of the spatial distribution of light intensity across the array[1]. Each of these discrete time analog outputs is commonly known as a pixel. Pixel sizes

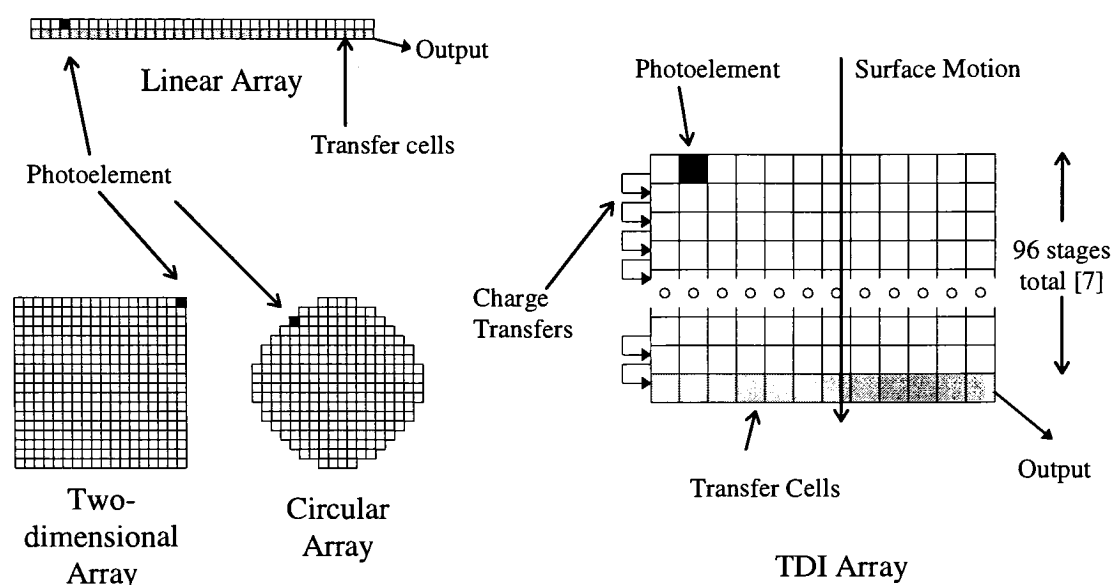
range from 7 by 7 microns to as large as 50 by 50 microns, with the typical size around 13 to 14 microns square.



**Figure 1-4: An Operational Summary of a CCD Sensor**

CCD sensors exist in several forms, as shown in Figure 1-5. Two-dimensional arrays are used in snapshot or frame-like cameras. Examples of these arrays are digital cameras used for photo-ID passes and camcorders. Circular arrays exist as well, but their use is not very applicable to web scanning. Linear arrays have a single row of photo elements, and obtain a two-dimensional scan by either moving, or by having a product move under them. Time delay integration (TDI) arrays appear as a combination of the linear array and two-dimensional array. The array is two-dimensional, but charge is transferred from one row of holding cells to the next row of holding cells when the transfer signal is received. The resulting charge is increased during the subsequent integration time in the new holding cell. When the final charge reaches the transfer cell row, it will act as if it has been integrating for multiple transfer periods, but still has a small, crisp pixel size. A typical TDI sensor may have 64 or 96 stages. Because the signal generated by the CCD sensor is proportional to its integration time, the TDI sensor allows a faster transfer signal rate while maintaining a significantly higher signal level. This allows the scanning of

web based products with reasonable light levels at the higher web speeds [7] (5,000 feet per minute).



**Figure 1-5: CCD Sensor Configurations**

## **1.5. Frame-Based Versus Line-Based Scanning**

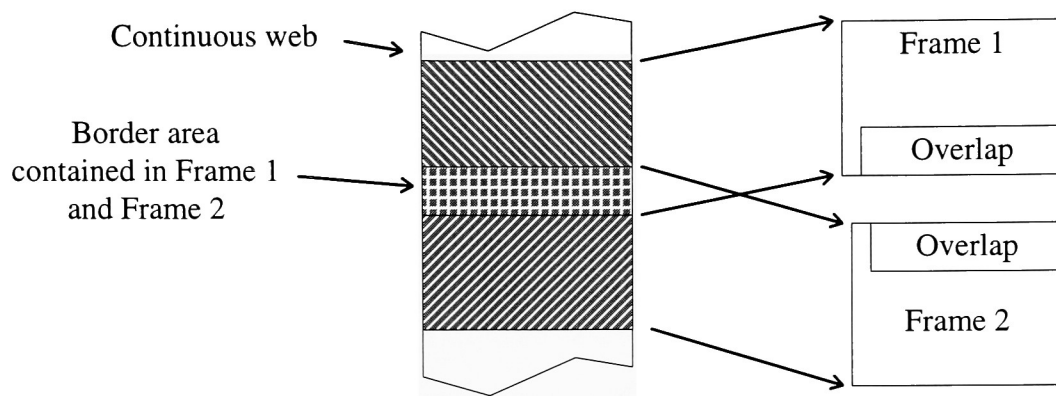
Two-dimensional CCD arrays, linear CCD arrays, and TDI arrays have been used in web scanning systems. Because the output of the linear and TDI arrays tend to act as a single continuous data stream, for this discussion they may be considered the same.

Many readily available image processing systems and board sets will allow a user to capture a frame of data from a camera. These are better known as frame grabber systems. Data can be fed to a frame grabber system by either a frame-based camera or a linear camera. The frame-based camera captures a two-dimensional image, and the frame grabber system reads the framed image from the camera. With linear cameras, the frame grabber system will capture a sequential series of one-dimensional scan lines and consider the captured lines as a frame. Because the linear camera is able to capture a continuous stream of one-dimensional image lines, this frame exists only in the frame grabber board.

Dedicated digital signal processing (DSP) boards can be plugged onto these frame grabbers to enhance the data, reduce the data, and provide analyzed outputs. But this requires time to process the data after it has been captured. If the processing time is lower than the amount of time to capture a frame of data, it is possible to employ multiple image buffers and capture data into one buffer, while processing previously captured data in a different buffer. Yet another buffer will likely receive the processed data. Once the buffer which was capturing data image data has been filled, pointers to the image buffers

switch, allowing new image data to be captured into the previous processing source buffer.

Several problems exist with dividing a web into discrete portions as required by a frame grabber system. Reference Figure 1-6. First, the web is artificially divided into discrete portions (which really do not exist). Many of the image enhancement features employed to identify imperfections work in two-dimensional space, relying on inputs from scan lines either above or below the current line. At boundary conditions of the frame, these bordering lines are artificially removed [7]. The result is a need to feed the image data to multiple image buffers simultaneously (at the end of one buffer, while at the start of the next) to remove the boundary conditions. Configuration must then keep track of how much of the image buffer is overlap. This also reduces the amount of processing time allowed on the image. Not all frame grabber systems provide this ability, and those that do tend to be the higher cost systems.

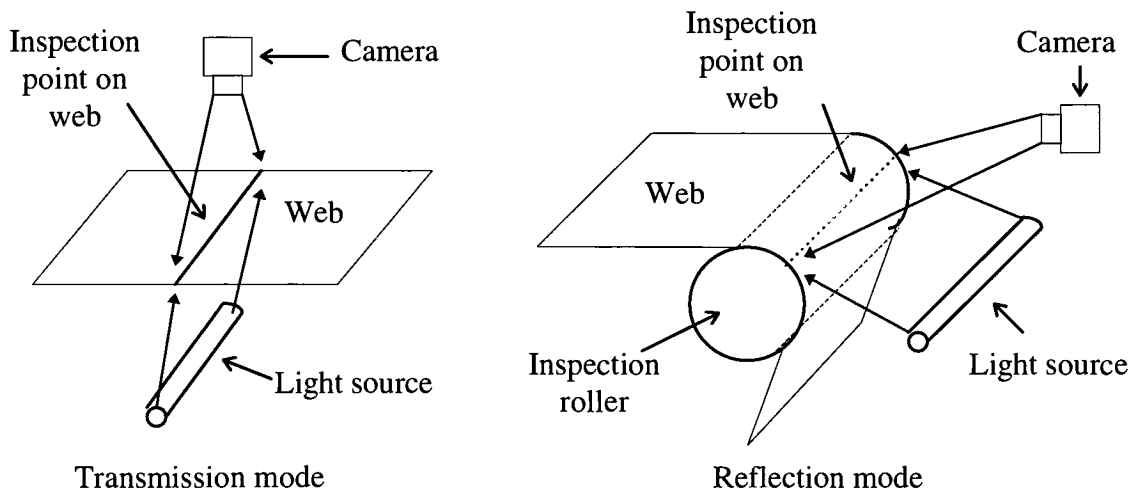


**Figure 1-6: Frame Overlap in Frame Grabbers**

The second issue is with processing times. Few frame-based systems have sufficient processing power to feed the captured image through image enhancement, defect identification, and connected component applications, in the time frame required. This is aggravated when the same processing portions of system are reused on each image processing step to reduce cost of the delivered frame grabber system. There are frame grabber systems that allow sufficient processing power to complete calculations in the allotted time frame, but these are the highest cost systems.

The frame concept is not well suited to the many web scanning processes [7]. To scan with a frame-based camera (employing a two-dimensional CCD array), snapshots of the web must be flashed with the equivalent of strobe lights. In transmission modes (shining light through a web to a sensor on the opposite side, as shown in Figure 1-7) this is somewhat possible if variations for light levels can be accounted for. In reflection mode (bouncing light off the product into a sensor) the product must be held taunt to obtain

accurate depth of field. This is difficult unless the product is held against a roller, which destroys the depth of field of the frame-based sensor on the upper and lower portions of the frame because of the roller curvature. This invalidates much of the use of frame-based cameras for web scanning.



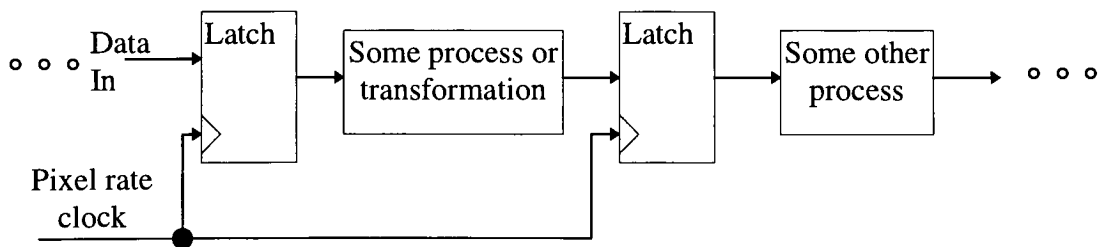
**Figure 1-7: Transmission and Reflection Modes**

With a linear camera, a line of light may continuously illuminate the product inspection area. It is also possible to employ a high-speed strobe light to illuminate the web, but still effectively in only a single line. Hence, fluorescent lights and “light tubes” can be employed, which have a lower instantaneous power surge to illuminate the web. Additionally, the inspection plane in a reflection mode becomes only one pixel high (described above to be on the order of 0.020 inches). This results in no limitation on depth of field issues around the curvature of the inspection roller. A TDI camera with 96 stages sees an imaging area of roughly two inches. A typical inspection roller has a 12.5 inch diameter. The resultant depth of field change between the ends of the roller image



plane and the center of the roller image plane is around 0.08 inches. This requirement is handled by most focusing systems, which support these web widths.

If the continuous stream of data, which is presented by a linear CCD array, is processed in a continual flow-through manner, issues identified with use of frames in web scanning can be avoided. As shown in Figure 1-8, a system implementing this scheme must be capable of performing any image enhancement with a single pass on the data. It is possible to pipeline many stages of image processing sequentially, and perform various operations on the data. It is not possible to allow iteration or looping in processing the image data in attempts to find or locate imperfections, because of the stream of data still to come from the array. Because this requires data to be processed in a deterministic manner with a guaranteed cycle time, the web scanning system becomes real-time.



**Figure 1-8: A Continual Flow-Through Process**

In [9], it has been argued employing linear CCD arrays and circuitry capable of receiving a continual stream of data makes systems more expensive than required. In the situation sighted, a web speed of only 600 feet per minute was employed with frame-based CCD sensors. This is insufficient speed to account for the most demanding applications. The strongest argument sighted against linear CCD arrays is the lack of modularity in systems processing data as a continual stream. Web scanning systems should, therefore, be designed to be modular both from the number of pixels, which can be scanned perspective, and the ability to match functions that are required to the task at hand. Providing more resolution than is required, or more functions than are needed, only make the system more expensive and more involved to train.

### **1.6. Production Line Data Acquisition**

Environmental conditions on a production line are frequently harsh. Acid baths, high humidity, and high temperatures are common place. To locate a complete web scanning system in this type of location requires extreme environmental chambers to house all the appropriate circuitry, display monitors, and user input devices. In order to significantly reduce the size of any environmental enclosure, data may be sampled through a camera and transmitted to the main processing engine. To minimize signal loss, the data should be digitized at or within the camera. For shorter cable routings (between 100 and 200 feet typically) coaxial or twisted shielded pair wires can be employed for transmitting this digitized data. For longer lengths, fiber optic cables must be used to eliminate the need for data repeaters in the transmission stream. The environmental enclosure then becomes only large enough to house the cameras and a transmitting board. The bulk of the

processing engine can be located away from the production line in a nearby “computer room”.

Digitization of data can employ several means of establishing the signal to digitize. The most common in modern CCD cameras is either a sample and hold circuit, or correlated double sampling. Sample and hold simply samples the analog signal output by the CCD array sensor at some appropriate point in the pixel output, and holds the voltage value such that the A/D circuit can digitize it. With correlated double sampling (CDS), circuitry recognizes there is noise present in the CCD array video stream caused by the reset signal provided to the array once for every pixel which is shifted out. Because the background signal (due to the reset signal) is slightly different for each pixel, two samples are taken from the CCD array output stream for each pixel (one at a dark reference level for the pixel, and one at the valid video reference level). The difference between these two signals is a true representation of the pixel response [10]. This difference signal is digitized, providing a higher fidelity representation of the signal response.

Video line transfer signals must be provided to linear CCD arrays to clock out each scan line of data. Origination of these signals stem from two philosophies: constant frequency and constant pixel size. With constant frequency, a crystal is divided down through counter circuitry to derive the required line transfer frequency. With constant pixel, an encoder is placed on one of the production line rollers (typically a roller with a driving motor to assure no slippage of the roller versus the product), which provides a pulse train

as the roller rotates. This pulse train is divided as necessary to provide an even multiple number of pulses per video line transfer signal. Hence, the size of the pixel in the down-web direction always remains constant to a given angle of rotation of the roller. TDI arrays require the use of constant pixel size to assure that the charge transfers move at the same speed as the web beneath them. The use of a constant pixel size allows a more definitive down-web direction, but also removes the constant illumination levels provided by a constant frequency scheme. Efforts must also be conducted to keep exposure levels entering the camera high enough to detect imperfections, but low enough to not saturate the CCD array.

No matter which video line transfer signal generation scheme is used, the web scanning system must track the constantly fluctuating web speed. Webs said to operate at 1,000 feet per minute usually fluctuate during operation by +/- 5% at best. Additionally, the web must accelerate up to that speed at web start, and decelerate down to a slower rate at web end. In order to properly place the position of an imperfection in the down-web direction under constant frequency operations, a scanning system needs information of this acceleration profile to relate to the time when a scan line was captured. No matter the video line transfer method, tracking of web speed is useful information for correlating the existence of imperfections if they only occur at certain speeds.

Pulse trains can come in two varieties: single ended and quadrature encoding. With single ended, the pulse simply shows rotation of the roller. With quadrature encoding, two pulse trains are provided with a 90 degree phase shift. In this manner, it is possible

to tell if a roller is moving in the forward or reverse direction. At slow web speeds, or use on rewinders where product may be moved back and forth, this allows the web tracking circuitry to accurately remember how far the web has actually moved. The pulse train is processed through some interface circuitry and eventually arrives at a controller board that handles the video line transfer signal generation, the tracking function, or both. Tracking function information is made available to a database system, and line transfer signals are distributed simultaneously to as many cameras as exist in the system. Cabling must be employed somewhere between the encoder and the database system, because the encoder is on the production line, and the database system is part of the main scanning system in a computer room.

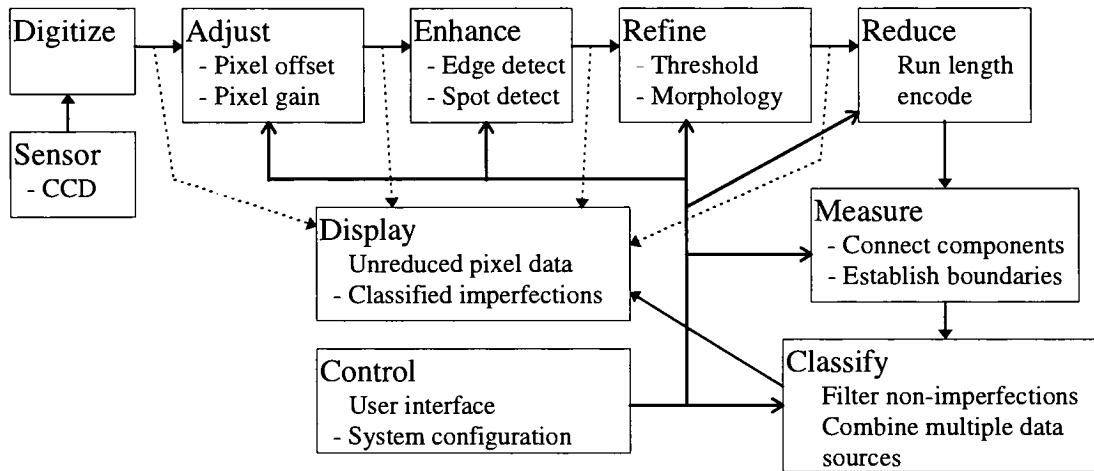
Responsitivity of any sensor tends to change slightly at different points along the sensor. CCD variations from pixel to pixel are usually only specified to within 5% of any neighboring pixel [10]. To compound the problem, light levels caused by motion of the web tend to continually fluctuate as well, especially in constant pixel mode. Product bends at different angles and moves in different patterns as it proceeds across the inspection point, making the optical signal constantly change. This results in changes to the scan line signal response, both roll by roll, and along the length of the web. This can be addressed by including the equivalent of a  $Y = mX + b$  formula after the sampling of CCD array outputs. More commonly, this formula is viewed as providing an offset to the received signal to remove DC components of the array, and then providing a gain to account for responsitivity differences. Most analog-to-digital conversion circuits provide a representation of this formula, in the form of an analog DC restore and gain operation,

which is active for all pixels in the array. Implementation of a pixel-by-pixel offset and gain is preferred in the digital realm to allow RAM or ROM tables to specify the offset or gain factor on a pixel-by-pixel address basis. The use of RAM tables allows a microprocessor to change these values as conditions change, such as a camera replacement or different product to scan. Further, if these RAM tables are implemented in a double banked configuration (ping-pong type memory) then values can be adjusted dynamically as a system scans to account for variations in light levels on the product.

### **1.7. Image Processing**

Many image enhancement processes convert the time domain information into the frequency domain, process the information, and provide resultant information in either the frequency or time domain as needed. This does not fit well for the general web scanning applications. A continuous stream of data does not transform well into the two-dimensional frequency domain without employing frame based systems. This has been identified above to be undesirable. Employing single-dimensional frequency analysis allows detection of continuous imperfections (such as streaks on paper or scratches on steel), but does not account for noncontinuous (including repeating roller mark) imperfections. The desire by web manufacturers to have a scanning system report the location of imperfections on the web in both a cross-web (width of the web) and down-web (length of the web) location, further places requirements on frequency domain operations to report information in the time domain.

By keeping image enhancements in the time domain, these concerns are avoided. A continuous stream of data can be fed to the processing circuitry. All reports of down-web and cross-web location can be accounted for by marking the number of scan lines passed and/or the number of pixels passed thus far in the current scan line. To accomplish this, significant processing power is required in the time domain. The result is a general system architecture as shown in Figure 1-9.



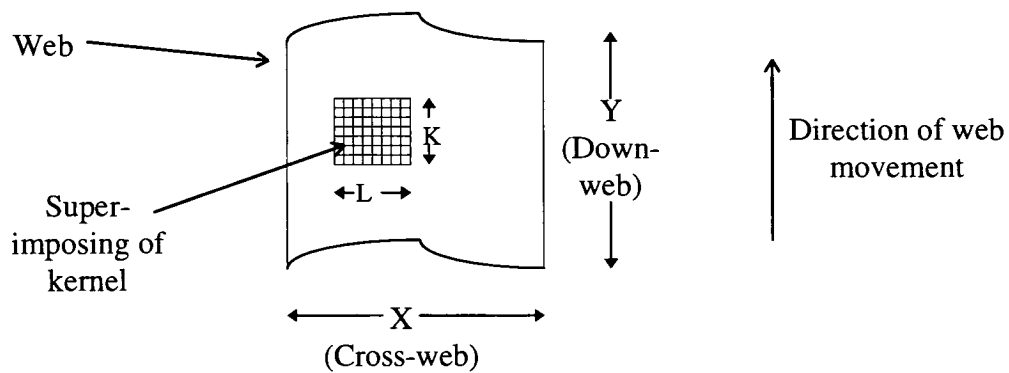
**Figure 1-9: A General Web Scanning Architecture**

The mainstay of most image processing systems in the time domain is convolution. Finite impulse response (FIR) convolution in the two-dimensional image domain is represented by the equation:

$$g(x, y) = h * p = \sum_{l=0}^{L-1} \sum_{k=0}^{K-1} h(l, k) p(x-l, y-k)$$

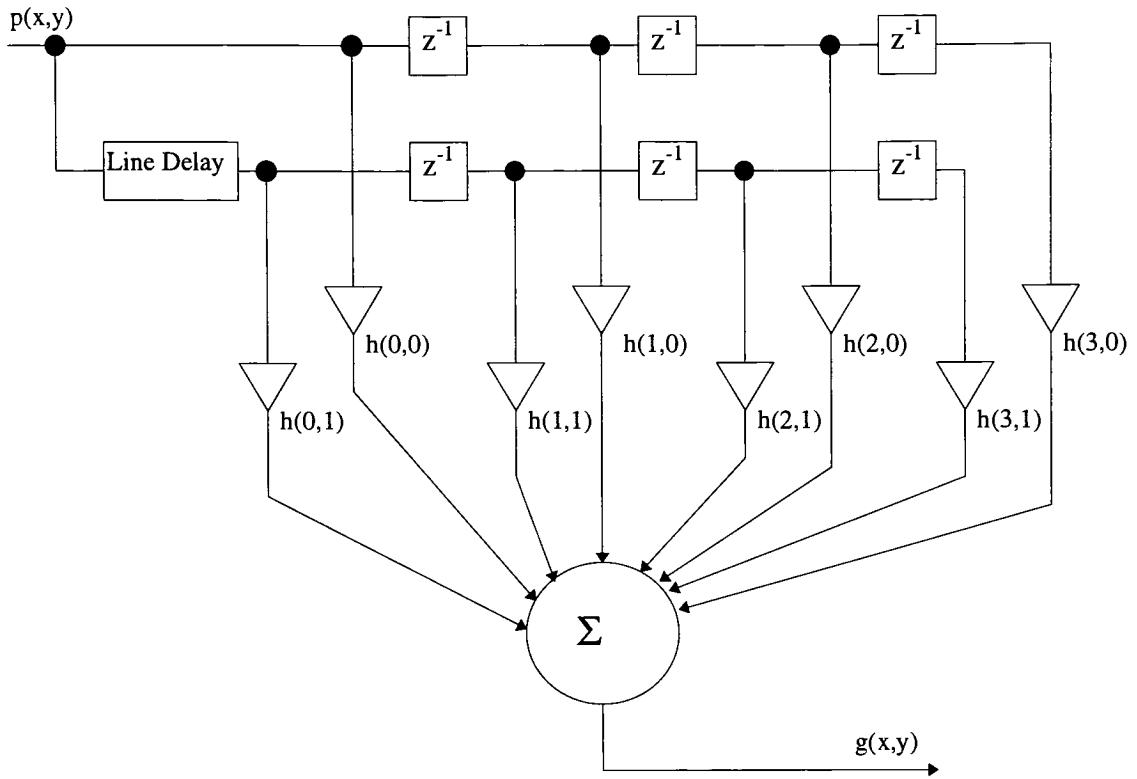
**Equation 1-1**

Where  $h$  represent the kernel,  $p$  represents the input image data, and  $g$  represents the resultant output value. The kernel has cross-web dimension  $L$ , and down-web dimension  $K$ , as shown in Figure 1-10. In the continuous flow-through architecture, this equation is effectively calculated once each time a new pixel is fed into the circuitry. The input image data ( $p$ ) is then shifted by one cross-web pixel in the positive direction at receipt of the next pixel. At completion of a scan line through the kernel, the cross-web values of ( $p$ ) are located in the same ( $l$ ) positions, but the ( $k$ ) positions have incremented by one. A generalized implementation of a FIR filter is provided in Figure 1-11.



**Figure 1-10: Orientation of Convolution Variables**





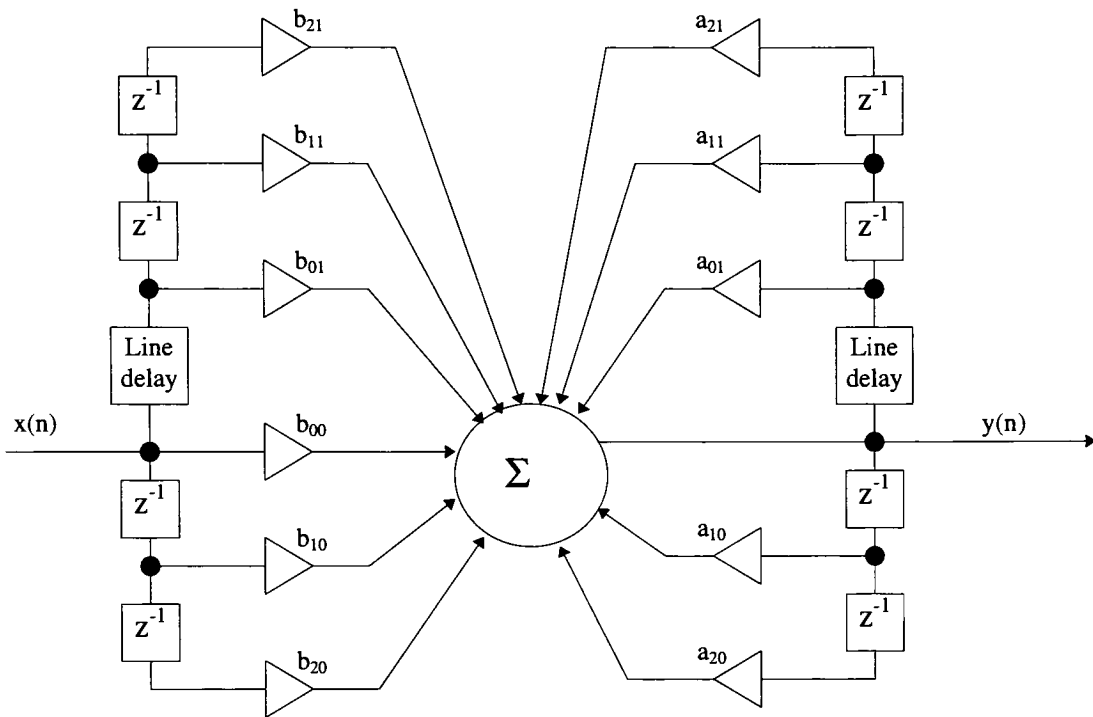
**Figure 1-11: A Generalized FIR Filter Implementation**

A generic infinite impulse response (IIR) convolution is represented by the equation

$$g(x, y) = \sum_{n=1}^N \sum_{m=0}^M a_{nm} g(x-n, y-m) + \sum_{m=1}^M a_{0m} g(x, y-m) + \sum_{l=0}^L \sum_{k=0}^K b_{lk} p(x-l, y-k)$$

**Equation 1-2**

Where  $a_{nm}$ ,  $a_{0m}$  and  $b_{lk}$  represent, filter tap coefficient values,  $p$  represents the input pixel data, and  $g$  represents the filter output value. This provides a generalized implementation shown in Figure 1-12.



**Figure 1-12: A Generalized IIR Filter Implementation**

Web scanning systems have been implemented using both FIR and IIR convolution filters. FIR filters are more common, because pixel data rates usually require the pipelining of multiplier and adder stages for successful implementation of a convolution circuit. By allowing multiple pipeline stages to arrive at the convolution output value, it is difficult to feed the output values back to the input of the circuit at scanning system pixel rates within one clock cycle, as is required by an IIR filter. IIR circuitry requirements are usually much lower than their FIR counterpart when providing the same specific frequency response, ordinarily requiring only a few multipliers and adders. Typical FIR convolution circuits considered for this implementation require from 9 to 64 multipliers, and from 8 to 63 adders.

Establishing FIR or IIR filter tap coefficients in web scanning relies on multiple variables. These include the type of product being scanned, the illumination technique (i.e., transmission or reflection), and the scanning geometry. Additionally, pixel size, and the type of imperfection to be detected (i.e., spots, water marks, diagonal scratches, etc.), all affect the required filter frequency response. On site experimentation is frequently required to fine tune filter coefficients to assure accurate imperfection detection. Thus, web scanning systems need to allow different filter frequency responses for each individual installation.

In [7], a web scanning system has been implemented that uses IIR filters for image enhancement. The circuit provides the application of the IIR filter in either the cross-web or down-web direction, but not both simultaneously. The result is the ability to apply the filter in two-dimensional space, while providing only a single dimensional filter. With these restrictions, it is usually more appropriate to apply this function for down-web filtering. The implemented circuitry allows the execution of a single pole IIR filter, represented by the formula:

$$U(n) = U(n-1) + \frac{(V(n) - U(n-1))}{T_c}$$

*Equation 1-3*

Where  $V(n)$  = the input data,  $U(n)$  = the present running sum of the IIR filter,  $U(n-1)$  = the running sum calculated from the last pixel, and  $T_c$  = a time constant to control the affect of  $V(n)$  on  $U(n)$ .  $T_c$  is required by the implementation to always be positive, and

must always be some even power of two. It is inferred the multiplier function is therefore carried out in a barrel shifter. The time required to implement the multiplication is significantly reduced, which resolves the issue of how to perform a multiply and an add in one pixel clock cycle as required for an IIR filter.

The demonstrated IIR filter implementation in [7] is a low-pass filter. The implementation also allows a high-pass filter function to be generated by delaying the original data ( $V(n)$ ) around the filter. The difference between the original data and the IIR filter data ( $U(n)$ ) is the high-pass filter output. Because most imperfections have a high-frequency component, a high-pass filter is generally employed for defect detection in web scanning systems.

Significant theory can be used for proper selection of FIR convolution kernels to be applied to find appropriate web scanning imperfections. This results in a vast array of potential kernels to employ. To avoid the need to identify all possible kernels and choose those which are most appropriate, the kernel coefficients should be made software programmable on web scanning hardware. FIR convolution chips exist, which provide the mathematical calculations required for an 8 column by 8 row kernel, operating at 20 megahertz (MHz), with 8 bit data and 8 bit coefficients [11].

Convolution outputs result in enhanced data with more bits per pixel than the initial pixel data. If 8-bit gray scale data is employed, and 8-bit coefficients are employed (including one bit as a sign bit), each multiply generates a 16-bit two's complement value. Adding

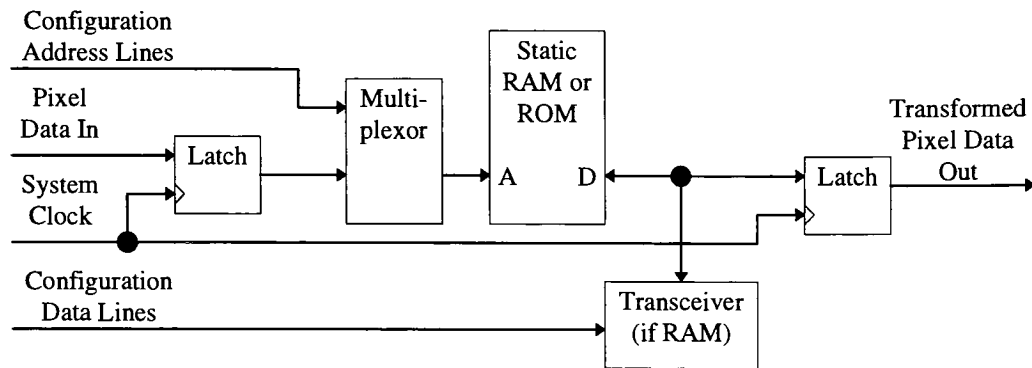
eight of these values together in the cross-web direction, and then eight of these partial sums in the down-web direction as required by an FIR filter, needs an additional six bits to fully represent the full sum. The result is a 22-bit value for each pixel. It is improbable all 22 bits will be required for the web scanning analysis; most web scanning systems allow selection of 8, 14, or 16 bits out of this demonstrated 22 bits by means of a data normalization circuit. Any value that cannot be represented in the smaller bit value, because of being too positive is clipped to the maximum positive value by a data truncation circuit. Similarly, a value that is too negative is represented as the most negative value the bits will allow. For example, 65,000 shows up in a 16-bit two's complement clip as 32,767; -65,000 arrives at -32,768. If this provides too many pixels that will be clipped, lower bits in the 22-bit sum can be ignored by choosing higher bits out of the initial sum with a right-shift operation before the clipping circuitry. Hence, if only 14 bits of data is usable by circuitry following the clipper, but the clip limits of -32,768 to 32,767 are desired for the input convolution data, the convolution output can be right shifted by two bits, and feed this result into the clipper. This has the effect of throwing away the lowest two convolution output bits.

The existence of the data truncation in web scanning systems introduces non-linearities. Without their use, the system suffers from either the possibility of pixel value overflows, insufficient signal level granularity, or extremely wide data bus requirements for functions following the filter stages. With their use, mathematical analysis of the system is complicated, and the possibility of linear analysis is removed. A properly tuned web scanning system will operate in the linear region under most web conditions. The system

only enters the non-linear region because of significant signal level differences created by imperfections, which is the point at which analysis would be most beneficial. As a result, on site verification of coefficient values is frequently required, because values calculated in a laboratory require adjustment once applied on site.

Clippers and barrel shifters work well to format input data into a different form. But their logic becomes complicated if the formats do not change at even power of two boundaries. For example, implementing a clipper function at a limit of 16,000 requires significantly more circuitry than 16,383. To make the clip value controllable by applications software at other than a bit boundary is not practical in discrete logic. Providing this functionality, and other more complex transformations, is best suited by the use of look-up tables (LUT).

Referring to Figure 1-13, a LUT places data on the address lines of a RAM or ROM device, and the information resulting on the RAM or ROM data lines is the transformed data. This is useful changing convolution outputs into thresholded values, moving from 16-bit linear mappings to 8-bit logarithmic mappings, and other equivalent functions. Once again, to keep the hardware as generic as possible, the use of RAMs is preferred to let the applications software mold the tables to whatever function is needed.



**Figure 1-13: A Generic Pipelined Look-Up Table**

A common use for LUTs is to transform multiple bit pixel data into binary, single-bit data. This allows changing 8, 14, 16, or other, bit data into one of two categories: 0 for below the tolerance threshold, and 1 for above the threshold. Hence, this process is frequently called thresholding. Because the values to be thresholded can be two's complement, the use of a LUT allows the tolerance level for positive values to be at a different absolute value than the negative values. For example, input values above +200, and below -100 can be considered 1, and between these two values to be 0's, or within tolerance. Resultant binary data can then be subjected to operations including morphology and feature extraction, which are discussed later.

For the web scanning application, the feature extraction process can make use of more information than just out of tolerance/background. It is useful to know if the deflection is positive or negative, or to what extent the deflection exceeded the threshold. In this situation, the information cannot be included in only one bit, so the implemented LUT tables produce multiple output bits. Systems have been generated allowing for two, four, and eight bit values to be passed on. To give this value a name, it will be called a "phase"

output, because it identifies the phase of an imperfection, in regards to positive/negative, or how far from the background value (which is always phase 0) the imperfection is.

A web scanning system needs to be modular, allowing both one camera for scanning a narrow web, and multiple cameras for scanning wider webs. Typically, data must also be received in the system from multiple scanning stations, i.e., double-sided scanning requires cameras on both sides of the web at different inspection rollers. When the data is received from multiple locations, there must be a means of combining the data from multiple sensors, and removing the overlap regions between the sensors. Because of edge effects on two-dimensional operations, such as convolution, it is desirable to minimize the edge effects at the cross-web boundary conditions by purposely over-scanning edge areas. It is then necessary to mask off the duplicated information when processing the data to avoid reporting an area of the web multiple times. An added benefit of the over-scanning/masking is to remove the end pixels from linear CCD arrays where the responsivity tends to drop off.

Most production lines will support multiple web widths. Webs also tend to weave from side to side during the manufacturing process. Scanning circuitry must be capable of tracking this weave properly, such that the area of the sensor input that should be ignored because of not being active product can be updated properly. The use of masking circuitry can handle the removing of unneeded sensor input, but requires the ability to update mask values as pixels flow-through the processing circuitry caused by the web weave.



Morphology can be employed to aid in the detection of imperfections. Detailed explanations of the operations of morphology can be found in [12] and [18]. The two basic operations employed in morphology are erosion and dilation. Erosion is demonstrated by the equation

$$A \ominus B = \bigcap_{b \in B} A + b$$

***Equation 1-4***

Dilation is represented by the equation

$$A \oplus B = \bigcup_{b \in B} A + b$$

***Equation 1-5***

The combination of pixels within a close neighborhood into one imperfection can be performed by conducting a dilation (to group local pixels together) and an erosion (to restore the outside boundaries of the overall group). Establishment of the center line parameters of an imperfection can also be performed, which allows the training of the web scanning system to be based on a significant subset of the characteristics that a particular imperfection may take on. Circuitry exists that can perform these functions on binary data [11]. Extensions to the concept can be generated allowing use with multiphase data (more than a 0 and a 1 for phase values), assuming imperfections with a higher phase number are more important than lower phase numbers. In this manner, dilations will overwrite a pixel with a lower phase number with a higher phase number. Erosions simply shrink all outside boundaries inwards uniformly across all phase

numbers. These can be effected through the use of a Rank Value Filter [11], which is represented by the equation

$$y(n) = SELECT\{j\} \left( \bigvee_{i=0}^k SORT(x(i)) \right)$$

***Equation 1-6***

Where *SORT* sorts the input  $x(i)$  values into increasing magnitude, and *SELECT* picks the  $j$ th element from the provided list of values. Multiphase dilations set  $j$  to the largest magnitude of the list, and erosions choose the smallest element of the list.

The ability to perform morphological functions in a continuous flow-through architecture requires the functions to be completed in a single pass through the circuitry. The ability to perform a dilation followed by an erosion, therefore, requires multiple pipelined stages of morphology circuitry.

## **1.8. Imperfection Extraction**

Data reduction is employed in web scanning systems. It is required to allow less data to be transported or transmitted from one location to another, and to reduce the overall data rate for presentation to a higher level of processing. This can be viewed from multiple contexts. For instance, raw image data coming from a sensor is reduced using a lossless compression scheme. This reduces the overall bandwidth on the transmission media between the production line and the main scanning system. Data is then uncompressed at the receiving end prior to image processing steps.

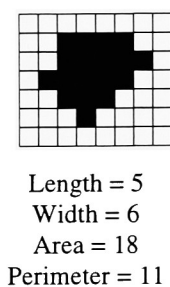
In another context, the presentation of classified imperfections to the controlling host computer has undergone a significant data reduction; the report need only indicate that an imperfection occurred, what it is, and where. This represents a significant savings in bandwidth between the processing engine and the host. A form of this technique was used in [2] to significantly simplify the complexity of a scanning system. Specifically, circuitry was provided within the camera enclosure to extract pixels which were considered as non-background pixels, and report them to the controlling host. A proprietary algorithm implemented in a reprogrammable logic array, detects pixels that lie outside an allowed pixel intensity window (i.e., are brighter or darker than the allowed pixel intensity). These pixels, plus an indication of where the pixels were located cross-web and down-web are reported. Pixels within the intensity window are discarded at the camera, and those which are reported by the camera pass through one common camera interface board. This allows up to twelve cameras to be handled by a single camera interface board, and a 80386 based PC. The data rate for each individual camera is fairly low because of the employed reduction techniques.

Additional methods exist to allow for the reduction of data and/or the data rate. Run length encoding counts the number of pixels in a row at a particular phase, and reports the phase number and number of pixels. This information is then frequently presented to microprocessors, since they typically can handle the resulting data rate. This information is then used to generate a connected components analysis. Another means of data reduction is to group raw data into a series of bins through dedicated circuitry for histogramming capabilities, to allow trend analysis.

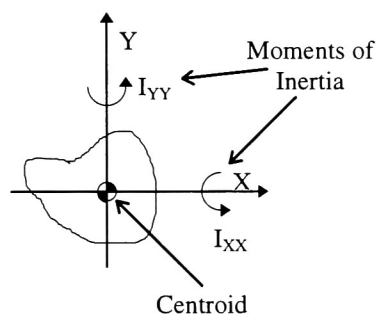
As technology advances, methods are developed to perform the functions microprocessors have done in dedicated circuitry which operates at pixel throughput rates. Faster microprocessors also alleviate the need for massive data reduction needs. But eventually, data must be reduced to show only the detected imperfections, and filter out all other information.

Connected components analysis deals with analyzing the phase information received from the data stream, and extracting features by following the connected phases. Measurements of the feature are conducted while each feature is extracted. This process is further complicated by processing the data in a scan line by scan line mode, since multiple features can exist on the same line. The algorithm must therefore allow multiple features to be active at the same time, and keep building features in parallel. Some features will likely be ending as others start. Minimally, a feature's area, length, width, and phase are computed (what will be referred to here as a feature's "normal" measurements). As classification needs increase, additional measurements must be performed, many of which are derived from other lesser measurements. The following is a list of measurements that have been employed by various connected components implementations, and cannot be derived from lesser measurements [13]. Examples of these measurements are contained in Figure 1-14.

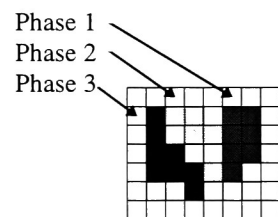
- Perimeter (the number of boundary pixels of the feature)
- Centroid
- Moments of inertia
- Density (number of on-phase pixels versus off-phase pixels due to holes in the blob)
- Ratio of phases (percentage of blob at each given phase)
- Maximum number of consecutive non-zero phases scanned in the horizontal direction
- Maximum number of consecutive non-zero phases scanned in the vertical direction
- Average number of consecutive non-zero phases scanned in the horizontal direction
- Average number of consecutive non-zero phases scanned in the vertical direction
- Orientation (the direction of the feature's major axis).
- Length of the major axis
- Length of the minor axis



Feature A



Feature B

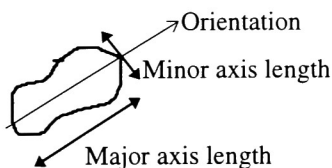


Density = 20/23  
Phase 1 ratio = 7/20  
Phase 2 ratio = 5/20  
Phase 3 ratio = 8/20  
Max Non-zero Horz = 5  
Max Non-zero Vert = 4

Horz Non-zero phase segments:  
3, 2, 1, 4, 2, 2, 5, 1;  
Average Non-zero Horz = 20/8

Vert Non-zero phase segments:  
4, 1, 2, 2, 2, 1, 1, 4, 3;  
Average Non-zero Vert = 20/9

Feature C



Feature D

**Figure 1-14: Example Feature Measurements**

Means have been proposed to perform some of these measurements in dedicated hardware [8]. The norm is to conduct these measurements in software because of the complexity of the algorithms, and the amount of circuitry required to successfully implement normal measurements. The more complicated the measurements, the longer the processing time to find the measurements and/or the more hardware required to perform the measurements. Because of board space and time constraints, most scanning systems calculate only the normal measurements and a few token measurements which also require very little processing time [8].

Classification employs the outputs of connected component operations, and potentially combines information from multiple sources together to accurately classify imperfections. Multiple features in close proximity can be evaluated to aid in establishing identity. Periodicity of a feature has impact on its classification, so the repeating nature of the feature is tracked. Accurate down-web placement is employed, requiring feedback from the web tracking subsystem in constant frequency mode. If double-sided scanning, or a combination of transmission and reflection mode configurations, are employed, the reporting of features from the multiple scanning heads can be combined together in issuing a decision. The same data can be processed with multiple image processing paths, allowing different enhancements to be performed on the data. Information from the multiple paths can therefore be merged in some manner as well.

Classification methods are frequently implemented in software within the web scanning system. To date, most of those which have been implemented in scanning systems have

been rule based. With the variety of inputs that are possible for leading to the classification of imperfections, and the need for faster classification times, this appears as an excellent application for neural networks. In [16], a neural network implemented in software is used to assist in the detection of imperfections in the manufacturing of Lace. The rudimentary classifier receives 75 input values, and provides a single output: imperfection, or non-imperfection. For the general web scanning case, an individual output would be desired for each different kind of imperfection which could be classified. The cited article states the demonstrated system is lacking in processing power to handle the rudimentary classification which has been implemented. Application to the general web scanning classification problem will require even more computational capabilities. Fortunately, work is underway on establishing hardware accelerators for neural networks, which should assist in their use within general web scanning systems.

## **1.9. Verification Aids**

Both in the training phase of a web scanning system, and in the general use of a web scanning system, production line personnel require visual information about imperfections the system has detected. Results after classification and presentation to a controlling host computer are not part of the unique scanning system hardware, and are consequently not detailed here. Viewing of image data prior to data reduction does have an impact on the web scanning system hardware architecture, because means must be provided to capture the data from specialized processing boards.

It may be desirable to capture image data for viewing after any image processing step has been applied within the web scanning system until the data has been reduced or compacted. The fundamentally simple task of capturing this data is complicated when it is required to capture data from multiple sources onto the same display subsystem. This is useful in web scanning systems, both to reduce system cost, and to allow display of pre-enhanced and enhanced data on the display simultaneously. This allows a user to verify what an imperfection looks like before enhancement, and also verify if enhancement has performed as expected. It is also possible multiple image processing paths exist in the web scanning system on wider webs due to the modularity of the system. Routing of data from multiple locations within the scanning system hardware to the display subsystem must be expected.

Consider a web scanning system implemented with 2048 pixels cross-web, and 20 million pixels per second scanning rate. This results in near 10,000 lines per second.



This rate is clearly too fast for a human operator to view data on a display monitor. Data must therefore be presented in frames to the user. These frames are displayed on monitors which top out at a resolution of 1600 x 1280, and a more commonly provide a resolution of 1024 x 768. Using the high-end case, several issues can be observed: First, there are insufficient cross-web pixels in the monitor (1600) to display all scanning system pixels (2048). This problem is aggravated as the number of pixels cross-web is increased. Sampling of pixels in the cross-web direction must therefore be provided. Second, each frame will be presented to the user for less than 0.13 seconds without some form of frame sampling. A human operator will be unable to gather useful information from a snapshot with a display time of less than a second. Circuitry is therefore required to sample the captured data, and present it as a display to the user at a reasonable frame rate.

Transitioning the system flow-through data into frames would appear as an argument for frame-based scanning systems. Implementing the less than 100% display rate requirement on a frame-based system requires capabilities most frame-based systems do not possess. These systems must usually display every frame they capture on their display, or none at all, without continual configuration and control by a host microprocessor. This is usable, but consumes more interrupts and processor cycles, which are needed for the primary feature extraction task.

A strong argument for a display subsystem is to allow visual verification of what an imperfection looks like when the web scanning system finds it. Simply providing a

snapshot capability requires probabilities and fate to be on the side of the user for capturing a particular imperfection on the display. For example, if one out of six frames is displayed to the user, there is a one in six chance any particular imperfection will show up on the display. A more deterministic approach provides for a continual sampling of data at the display subsystem, and a means of feeding back information from the connected components and/or classification portion of the system to the display subsystem. Sufficient buffer space memory must then be provided in the display subsystem to allow for the processing time of the analysis section to find the imperfection.

## **2. Implementation**

The Veredus Quality Control System (Veredus) was started at Eastman Kodak Company in 1984. The concept was developed to fill a need at Kodak to scan internally manufactured products, while taking advantage of CCD linear sensor technology. These sensors offered financial savings for the implementation of the scanning head. All other production line web scanning systems in use at Kodak to that point employed more expensive flying spot lasers for the scanning heads. As the technology development continued, applications external to Kodak were identified, and the system was sold and installed external to Kodak. The Veredus has been applied scanning paper, film (both base and sensitized), tin plate steel, cold roll steel, aluminum, copier base materials, and titanium rods.

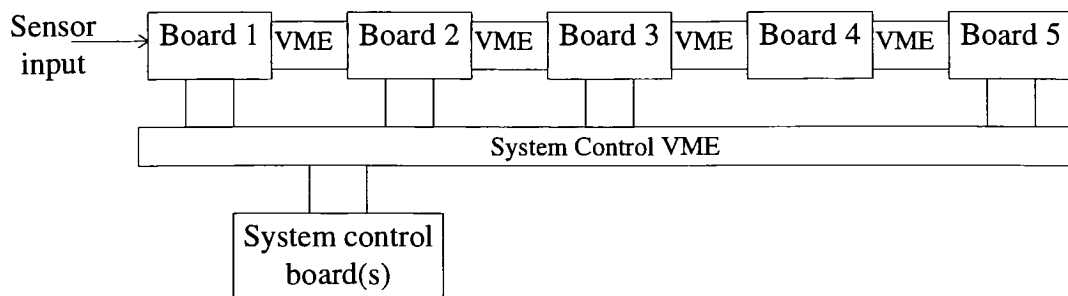
The Veredus is intended for use in industrial, high-end applications. It has been used at web speeds up to 5,000 feet per minute, at pixel sizes of 0.020 inches. The system modularity also allows use in applications with web widths of 200 inches, running at speeds of 1,000 feet per minute. The harsh atmospheric conditions encountered on the production lines have driven the system design, providing application to scanning situations which allow few competitors.

Technological advances since 1984 have resulted in continual improvements to the Veredus. Backward compatibility has always been a primary concern, requiring both hardware and software advancements to be applicable to previously installed Veredus

systems. The hardware requirement is to find ways to place new functions within board input and output interfaces defined by earlier designs. If a new intraboard interface protocol is required for a board or series of boards, provisions must be made to support any old, previously existing protocols as well. As required, a design project includes an interface board to allow a bridge from an old protocol to the new one. The design project also includes an analysis of how to place the new boards in a previously existing chassis, with a finite number of board slots available.

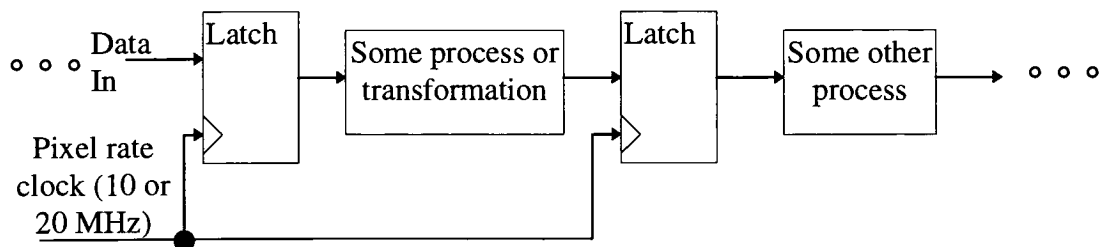
## 2.1. Architectural Overview

The basic architectural construction of the Veredus revolves around a signal or image processing pipeline, as shown in Figure 2-1. More commonly referred to as a “pipe”, the image processing pipeline consists of four and seven boards, interconnected by VME P1 type backplanes. Boards can be shared between pipes in some configurations. Additional circuit boards can be included by connections other than the VME P1 type backplanes, such as analog backplanes, cables, and daughter board connections.



**Figure 2-1: Backplane Data Flow Configuration**

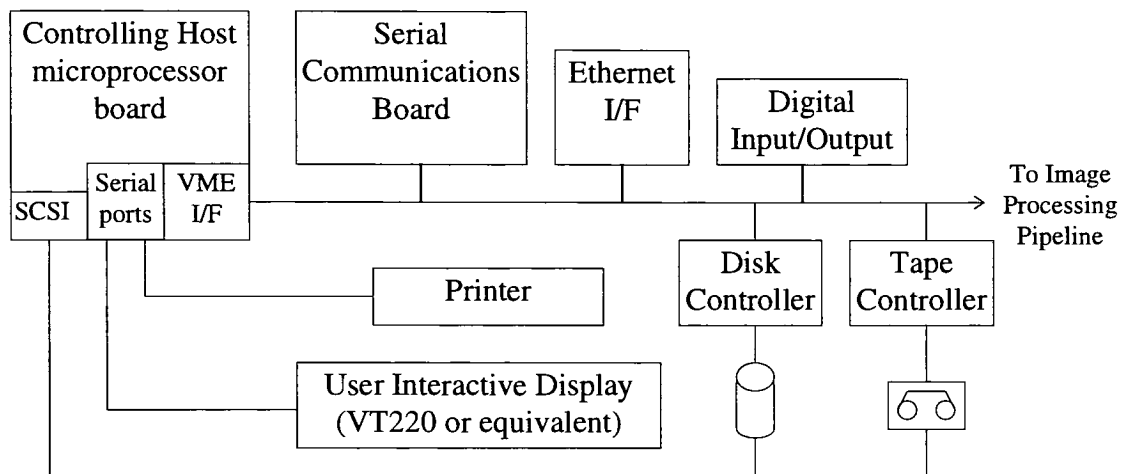
Data flows through each pipe in a continual flow-through process, as shown in Figure 2-2. Older boards in the system process information at a maximum data rate of 10 million samples per second. More recent designs allow data to flow at 20 million samples per second. This results in a pipelined process on each circuit board, where the time to process a pixel in each stage ranges from 100 nanoseconds (ns) down to 50 ns. At the completion of a pipeline stage, data is shifted into the next pipeline stage to continue processing, while new data arrives in the present stage to be processed as well. This results in a continual stream of pixel data from sensors entering the pipeline boards, and a continual stream of connected component output (or feature vector) data leaving the pipeline. The feature vector data bandwidth is significantly lower than the initial sensor pixel data, and is bursty in nature, i.e., data will exist for a period of time when imperfections are found, and then none will exist when there are no imperfections.



**Figure 2-2: Continual Flow-Through Process**

Feature vector data exiting from the pipe is used by controlling microprocessors to report the type and location of imperfections found. The reporting is carried out on graphical display monitors, printed reports, and communications networks (RS-232 and ethernet), as Figure 2-3 demonstrates. In order to assure this feature data is properly received and handled by the end user, feature vector information is stored to disk for later retrieval.

Requests for the scanned product imperfections are therefore available for query at later times through the interactive display of the Veredus, or through communications networks. These data files can then be written off to a streaming tape backup, and deleted, or simply deleted if the user so chooses.



**Figure 2-3: Interconnection of Host Rack**

Wherever possible, commercially available circuit boards are purchased for use in the Veredus system. Controlling host microprocessor boards, disk controllers, tape controller, serial communications boards, and data input/output boards are acquired in this manner. Bus repeater and bus bridging board sets, such as VME bus to S-bus, are also acquired this way.

## **2.2. Interboard Connections**

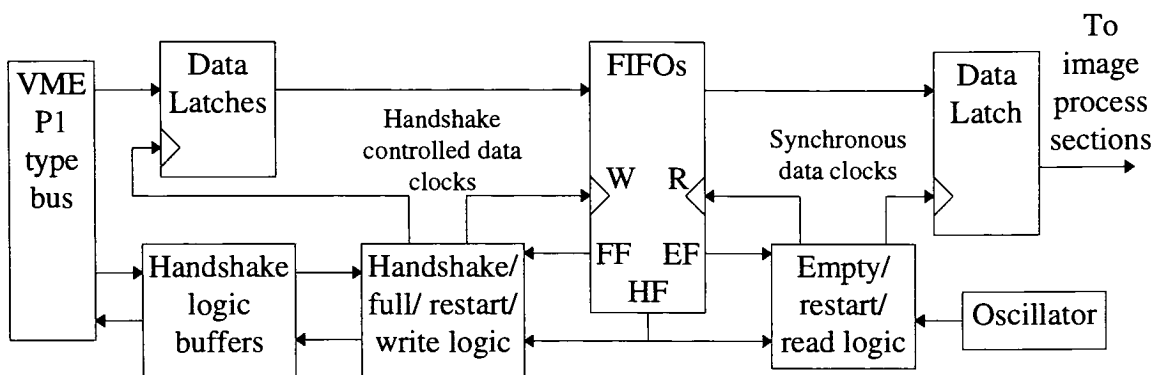
Pixel data transactions on the VME P1 type backplanes rely on typically having only one master on each backplane, and potentially multiple receivers, or slaves. This results in the removal of standard VME bus arbitration and bus controller logic, significantly simplifying the interface logic on each board. Furthermore, bus handshaking logic flow is maintained, but timing requirements are shortened to increase throughput on the backplane. Transactions typically operate at up to 10 million transfers per second on older designs, and have been observed on the most recently designed boards to work reliably at over 15 million transfers per second. VME P1 type backplanes officially support 24 address lines and 16 data lines. Because most data transactions between newer pipeline boards require only eight bits per pixel represented, two pixels are transmitted on each bus cycle. This allows operations on newer boards to run at 20 MHz, while bus cycles are only required to operate at 10 MHz. In certain situations, such as when a board can emit data enhanced multiple ways, two data items are required for each pixel. In these cases, the A24 nature of the P1 type backplane is enhanced to allow 16 address lines to be employed for data between Veredus boards. These 16 lines are also divided into two parallel eight bit pixel values. In order to still allow use of the Veredus boards with commercially available VME boards, true address lines can be multiplexed on the lowest 16 address lines if needed.

The first series of pipeline boards developed for the Veredus directly coupled the operations on the boards to the VME bus cycles. While fundamentally this worked, it lock-stepped the entire system together, requiring input and output timing on all pipeline

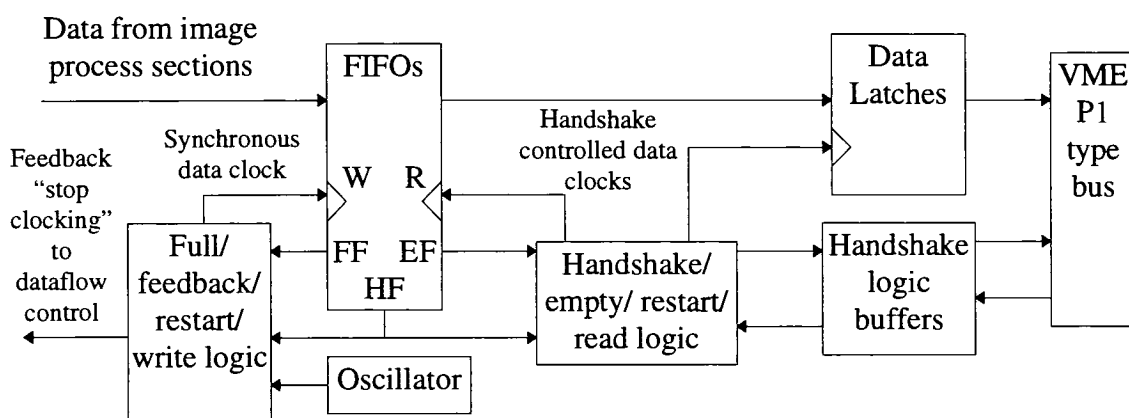
boards to be used for data flow clocking. System integration problems showed a timing change on one board would affect the proper operation of a systolic type operation on another. Newer boards alleviated this problem by placing First In First Out memories (FIFOs) between the VME bus and the processing circuitry on the board. Refer to Figure 2-4. Data into or out of the FIFOs from the VME bus side is controlled by the VME bus handshaking. Access on the processing side is controlled by an oscillator generated synchronous clock. The combination of FIFOs, and most recently pipelined, VME bus interface logic are termed the High Speed Slave (HSS) and High Speed Master(HSM) circuits.

The use of FIFOs require provisions for when the FIFOs become empty or full. Synchronous clocking must stop in the image processing section of a board when a HSS FIFO empties, just as handshaking will be suspended with no more data is available at the output of the HSM FIFO. The full case requires the HSS stop returning Data Transfer Acknowledge (DTACK) to the controlling Master, and a full condition in the HSM requires the synchronous clocking of data into the FIFO from the image processing section to cease. In all cases, the normal situation resumes once the FIFO presents a transition of the half-full flag.





Veredus High Speed Slave



Veredus High Speed Master

**Figure 2-4: High-Speed Slave/Master Configurations**

Veredus circuit boards have been designed to be dumb and fast. Control of the image processing boards has been provided by a controlling "Host" computer. Software operating on the Host has the responsibility to setup the pipeline boards as needed for the imaging applications. As a result, microprocessors have been minimized in the system to locations where "intelligence" is required. This conscious decision allows easier implementation of software control for the system, since there are fewer locations object code is operating. This decision complicates circuit board design, since many VME bus

interface circuits which are commercially available rely on the existence of a microprocessor on the designed board.

All board communications cannot occur over VME backplanes. Because the Veredus has allowed the decoupling of the sensor head from the main system, data must be received from cameras that are located external to the Veredus main processing system. Earlier systems allowed the receipt of analog data from the cameras, and digitized the data within the main system. This required control signals to be distributed within the main chassis between camera interface boards and the Analog-to-Digital Conversion Board which are not appropriate for the VME backplane. A 96-pin Analog Control Bus (ACB) matching the VME Euro-card form factor standard was designed to allow distribution of power (for both the boards and the cameras) and control signals. Digital outputs from the digital input/output board were then used to start and stop the A/D board, by connecting wires to the ACB. The maximum speed a camera could operate in this mode was limited to 10 million pixels per second, and a maximum distance of 100 feet between the camera and main system was obtainable.

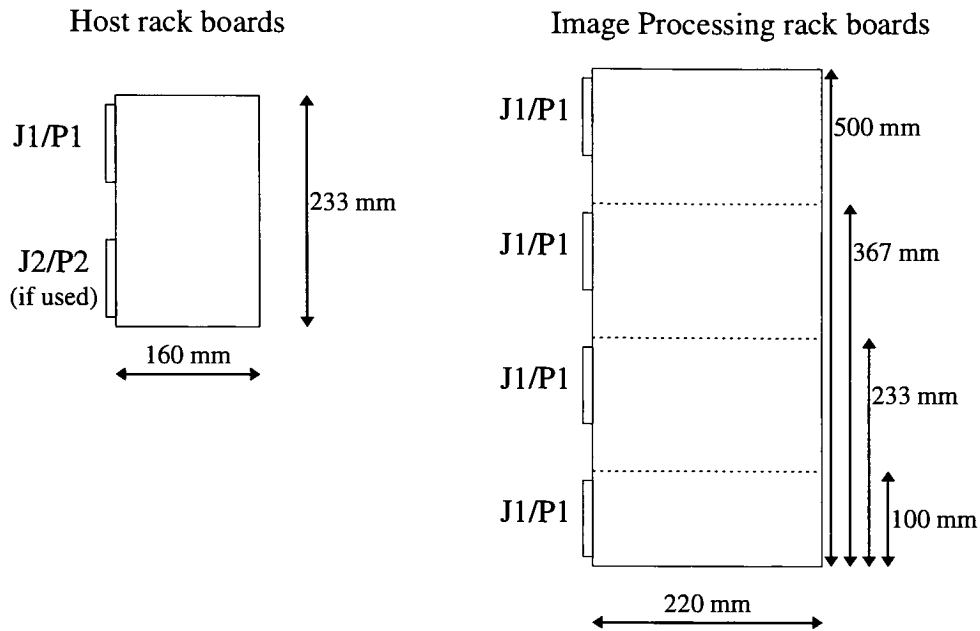
In later systems, the requirement for an ACB was removed by digitizing data at the cameras. Power supplies are now required at the camera, plus additional control signals must be sent from the main Veredus system to the controller boards to control the line transfer clock at the cameras. Serialized digitized data is sent back to the Veredus on either a pair of coaxial wires, or a fiber optic link for higher speed (above 10 million pixels per second) or longer distance (above 200 feet) applications. With a fiber optic

link, 12-bit data can be transmitted from the camera at data rates up to 20 million pixels per second, and distances up to 1 kilometer.

Cabling between the Veredus system and the production line is required to allow the Veredus to accurately track motion of the web. Rotary pulse encoder signals are buffered and sent in a quadrature format to a web tracking board in the Veredus. Contact closures on the production line are observed to allow the Veredus through digital input/output to tell when a roll has started, ended, or is known to be bad product (so that there is no reason to scan the product). The digital input/output can also send the status of the Veredus system, i.e., scanning, “in waste”, etc., back to the production line.

### **2.3. Board Format**

Circuit boards employed in the Veredus system fall into one of three categories: Use in the image processing rack, the host rack, and external to the system chassis. This last category allows for placement on a web production line, or on a wall. Boards employed in the image processing rack use a Euro-card standard dimension of 220 mm deep, while boards employed in the host rack use a Euro-card standard 160 mm deep format. See Figure 2-5. Because boards placed in the host rack always occupy the space of two VME bus connections (although both VME connectors may not be used), the height of these cards is a standard 233 mm. Boards placed in the image processing rack vary from 1, 2, 3, or 4 backplanes in vertical height, resulting in vertical dimensions of 100, 233, 367, and 500 mm.



**Figure 2-5: Veredus Board Form Factors**

Many configurations are possible with these varying dimensions. The ability to configure these boards in a modular format allows the Veredus to be customized for individual customer's needs. As a result, from one to eight image processing pipelines may be implemented in a Veredus system. Chassis and backplane configurations within the chassis vary, dependent upon the board set employed, the board connectivity required, and the size of the chassis.

## **2.4. Board Functions**

Although multiple generations of boards exist, they generally are referred to as one of several categories of boards. They are:

**Camera Interface:** Used to match the requirements of a particular sensor or camera to a generically designed analog-to-digital conversion/pixel adjustment/pixel statistical sampling board. Some boards allow the receipt of analog data from analog cameras; later boards form a matched set of boards, working with strictly digital data. These later boards have a transmitter in or near the camera, and a receiver located within the main system chassis. References here will call these interface (I/F), Digital Camera Transmitter (DCT), or Digital Camera Receiver (DCR) boards.

**Analog-to-Digital Conversion:** Receives data from a camera interface, digitizes (if not already digitized), performs pixel-by-pixel signal offset and gain corrections, and passes eight bit digitized data onto two-dimensional image enhancement boards. Earlier boards digitized data to 12 bits, and selected the most appropriate bits to be passed on as 8 bit data. Later boards actually receive eight or twelve bit digital data from interface boards, and have no indication of an Analog-to-Digital converter on them. Pixel-by-pixel offset and gain correction is still performed, and multiple digital interface boards can be multiplexed together. With historical references in place in documentation and software, these purely digital boards are

frequently referred to as A/D boards, but will be referred to here as a Digital Camera Multiplexor (DCM).

**Intelligent Filter:** Snapshots data from the A/D board outputs, and performs statistical analysis on pixel data trends. One-dimensional filtering is applied, and snapshots have been used to perform FFTs with an onboard DSP processor. A normal use of this board is to capture snapshots during run-time to account for light level variations on the moving product. The captured data is used to dynamically calculate new A/D board pixel-by-pixel gain values. These values are written to the A/D board, while the system is scanning, over a dedicated VME bus connected to the front of these two boards. Long running imperfections (such as streaks) are also detected in this process. In later designs, these functions are incorporated onto the strictly digital offset and gain adjustment board, or DCM. The Intelligent Filter Board will also be referenced as an IFB.

**Convolution:** Takes adjusted data from the A/D board, and performs a two-dimensional convolution of the data. On earlier designs, the output of a 5 x 8 kernel is directly transmitted to a thresholding and data reduction board, employing 16 bit per pixel accuracy. On later designs, up to four parallel 8 x 8 kernels can be employed, and thresholding is performed separately for each kernel. Results of the four parallel data streams are compressed into a mere 8 bits per pixel before sending to the data reduction board. Once again, even though many functions beyond convolution are performed, historically the board is only referred to as a Convolution board.

The earlier design is referred to as the Convolver 1.0 (CONV 1.0), and the newer design as Convolver 3.0 (CONV 3.0).

**Run Length Encoding:** Receives data from the Convolution board, thresholds it if needed to typically two bit phase values, and reduces the data through run length encoding. Earlier designs receive 16 bit data, run the data through threshold look-up-tables (LUTs), and generate a two bit phase value for each pixel. Run length encoding simply counts the number of sequential pixels at each phase. In later designs, the thresholding can be bypassed if the function is performed on the convolution board, but masking can be performed to remove edge effects due to convolution. Multiple run length encoding schemes are provided to allow use of the “old” scheme, or to generate data with reference to the two-dimensional space where each phase change occurs. The earlier design is known as RLE 1.0, while the newer design receives the name RLE 1.6.

**FIFO:** Provides a temporary buffer for run length encoded data to account for the bursty nature of the image data. As data is dumped into the board by the run length encoder, data is read out by a quad processor board performing connected components. When imperfections show up, more run length encodes are generated, and the quad processors cannot necessarily handle the instantaneous data rate. Providing the buffer storage allows the quad processors to catch-up after the imperfection has passed on the web. In later designs, a significant buffer

is provided on the Run Length Encoder, reducing the FIFO board to a bus-to-bus interconnection device with some additional buffer space.

**Quad processor board:** Provides four Motorola processors to perform general purpose processing on run length encoded data. While the intent is for general purpose processing, implemented routines are almost always a connected components algorithms. Resultant feature vectors are then passed up to a controlling Host microprocessor. Earlier board designs employed 68010 processors, and lack horsepower to perform much more than the connected components. Later designs employed 68020 processors, but the expense to port the connected components algorithm to the new architecture of the board was deemed economically unattractive. Work proceeded instead on implementation of a connectivity and measurement board, which implements the connected components algorithm with hardware accelerators and a Texas Instruments Digital Signal Processor. This design is functional, but has yet to be integrated to the rest of the Veredus system. Further references will use Quad or Quad-II to mean the 68010 based board, Quad-III refers to the 68020 design, and the connectivity and measurement becomes the C&M board.

**Gray Scale Display:** Capture and display 8-bit data on a video display monitor for use on site verifying proper operation of the scanning system. Earlier designs could only capture 8-bit data from a single A/D board output. Later designs allowed a slave board to be connected to either the A/D board or the Convolver board outputs, and

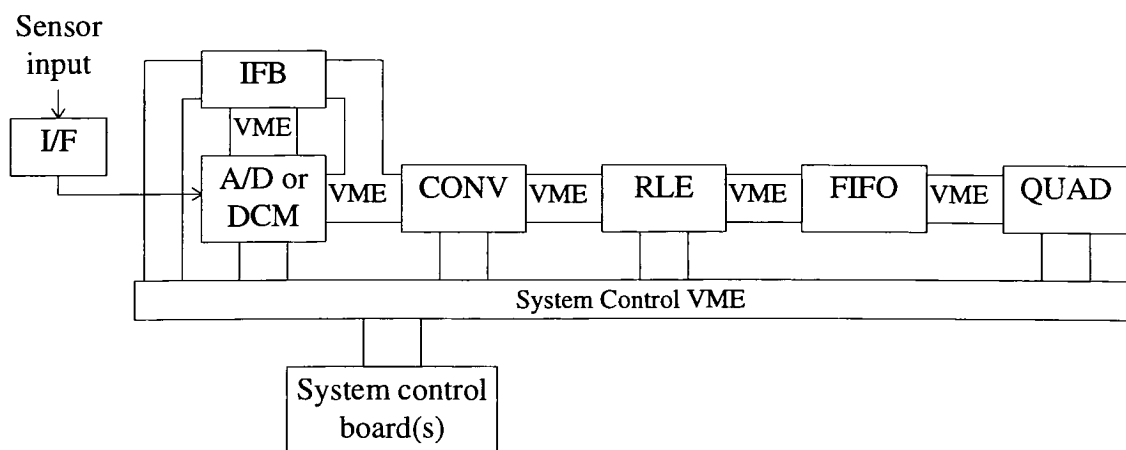


pass data through ribbon cables to the main display board. At the display board, data can be passed through a look-up table to map up to 16-bit resolution down to 8-bits for display. Up to four data sources can be fed to the display board, all of which can be displayed separately, or at the same time. Display monitors have transitioned from 512 x 512 resolution RS-170 compatible monitors to full 1024 x 1024 16 million color capabilities with the newer display monitors. LUTs exist to allow the gray scale information to be mapped to various colors on the latest designs, but the implemented use of this function adds little value to the presented display. The earlier design is identified as a GRAY board, while the newer interface board and display board combination are referred to as an IC and GRAY 3.0 board, respectfully.

Web tracking: Allows receipt of buffered rotary encoder quadrature encoded signals to track proper motion of the web on the production line. Earlier designs simply received the encoder signals, put a count of the pulses in a register available to the host, and returned a single ended line transfer pulse that was some integer multiple of the rotary encoder pulses. Later designs allowed a hysteresis and jitter control on the received encoder pulses, to account for noise on the rotary encoder. An oscillator controlled line transfer pulse for constant frequency use is also provided. The later design drives four parallel differentially driven start of scan signals. Both boards will arrive on diagrams as WTRK boards, but a reference to Web Track 2.0 indicates strictly the later design.

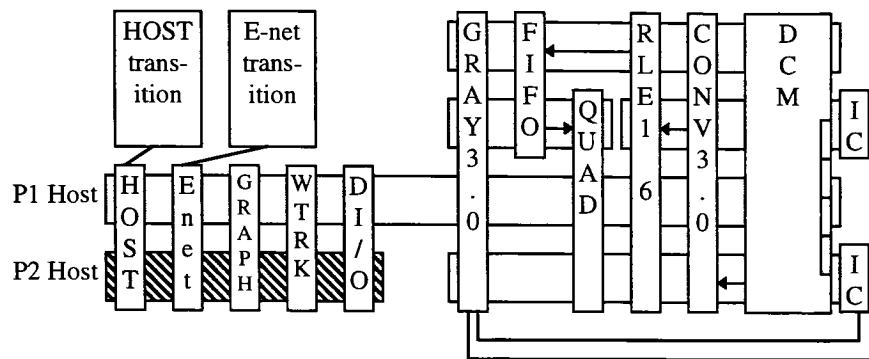
Hosts boards: Commercially available circuit boards, providing functions such as a controlling microprocessor board with a Motorola processor on it, known as “the Host” microprocessor. Disk and tape drive interfaces, ethernet connections (E-net), serial port connections, digital input/output (DI/O), graphical display monitor drivers (Graph), and bus repeaters (RPTR)/bridges also fall into this category. These boards have been updated as they have become available from manufacturers, and been supported by employed operating systems.

Backplane configurations for a “Veredus 250” with one pipeline capability, and a “Veredus 1000” with up to four pipelines, establish the connectivity of these boards. These are shown in Figure 2-6 through Figure 2-9. These configurations provide examples of how a later generation of boards significantly reduced the complexity of a system to provide the equivalent, or more, scanning functionality.



**Figure 2-6: Veredus Pipeline Block Diagram**





Host and Ethernet processors boards have their 2-slot transition boards still connected by ribbon cables as performed in the Veredus 1000. No backplanes are needed to support these boards.

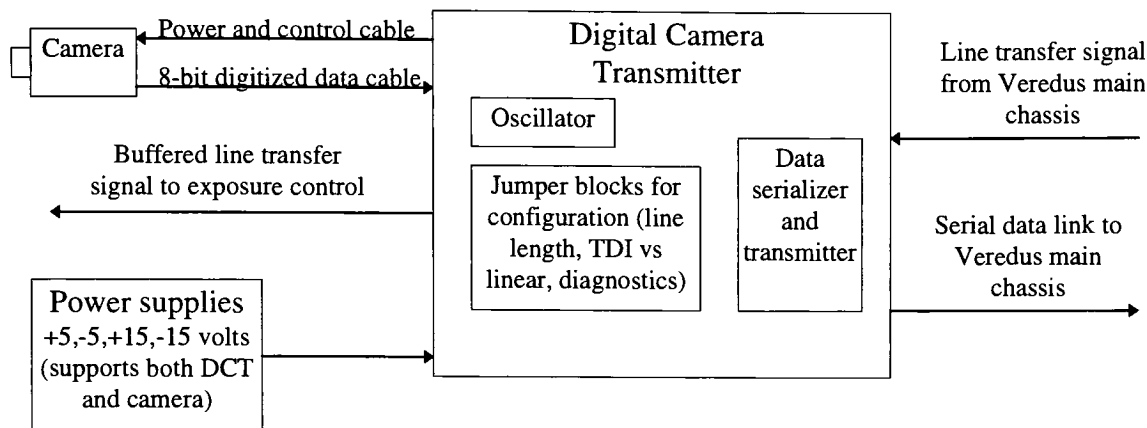
Functionality in this Veredus 250 is equivalent to that of the Veredus 1000 in Figure 2-8, but can operate at twice the speed (20 MHz).

**Figure 2-9: A Veredus 250 Configuration with Newer Generation Boards**

## **2.5. Data Acquisition**

The Veredus receives data collected from CCD sensors. Both linear and TDI arrays are employed for this purpose, with pixel widths of 512, 1024, 2048, and 4096 pixels per sensor. Analog data is clocked from these arrays with circuitry internal to a camera placed on the production line. On Veredus systems currently under production, data is then digitized in one of two forms. In the lower cost case, cameras purchased external to Veredus sample and hold the analog data, then digitize it to eight bits per pixel at data rates up to 15 MHz. For high performance requirements, a Veredus designed camera employs correlated double sampling on the output from a linear CCD array to allow 12 bits per pixel digitization, at data rates up to 20 MHz.

CCD array line transfer signals are provided from the main Veredus chassis via cabling. This differentially driven signal is routed through either a power distribution assembly for Veredus designed cameras, or a Digital Camera Transmitter board for externally purchased cameras. In the case of externally purchased cameras, this allows the DCT to gather all signals needed to control the externally purchased camera, as shown in Figure 2-10. This includes the routing of power from power supplies, and the internal generation of pixel clocks. Cabling between the DCT and the camera is therefore two simple straight through cables, i.e., one connector on each end of the cable, and the same pin-out on each end of the cable. This allows the use of commercially available cables, rather than having specialized cable assemblies for gathering signals from multiple sources in the cable itself. Differentially driven data is received from the camera on one of these two cables by the DCT, and formatted for transmission to the main Veredus chassis.

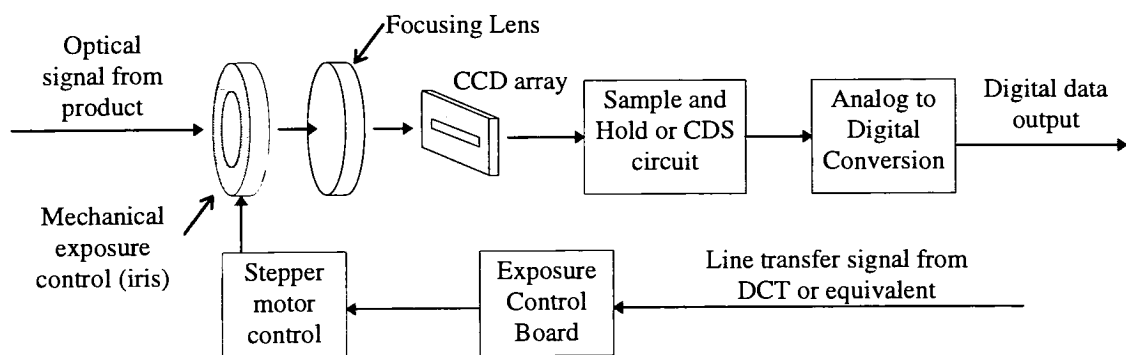


**Figure 2-10: Digital Camera Transmitter Connections**

The Veredus designed cameras receive power and the line transfer signal over a single straight through cable as well, from the power distribution assembly. All pixel clocks are locally generated within the Veredus designed cameras, and the resulting output data is formatted for transmission to the main Veredus chassis.

The line transfer signal originates at the Web Track 2.0 board in the main Veredus chassis. By keeping generation of these line transfer pulses in the main Veredus system, all cameras within the system have a common, synchronized scanning rate. Software control of the process is also achieved, because controlling software has one common point within the system to access which controls the line transfer generation. Selection of the transfer mode between constant frequency (based on an oscillator output), or constant down-web pixel (based on the movement of the web) is provided by configuring the Web Track 2.0 via Host microprocessor accessible registers. Appropriate frequency and/or rotary encoder pulses per line transfer are also controlled in this manner.

To accommodate light level fluctuations in constant pixel mode, the start of scan signal is provided to an automatic exposure control circuit. Reference Figure 2-11. The time between line transfer signals is sampled, and employed to move mechanical assemblies via stepper motors. The stepper motors decrease the amount of light incident on the optical lensing and array assembly at slower web speeds. This subsystem is configured such that the mechanical assembly allows the maximum amount of light (signal) through at the maximum web speed (shortest time between transfer pulses). As web speeds then decrease, the time between transfer pulses increase to maintain a constant down-web pixel, which allows additional time for charge to gather in the CCD array. The mechanical mechanism is adjusted to decrease the amount of incident light to account for the longer integration time.



**Figure 2-11: Camera Subsystem with Exposure Control**

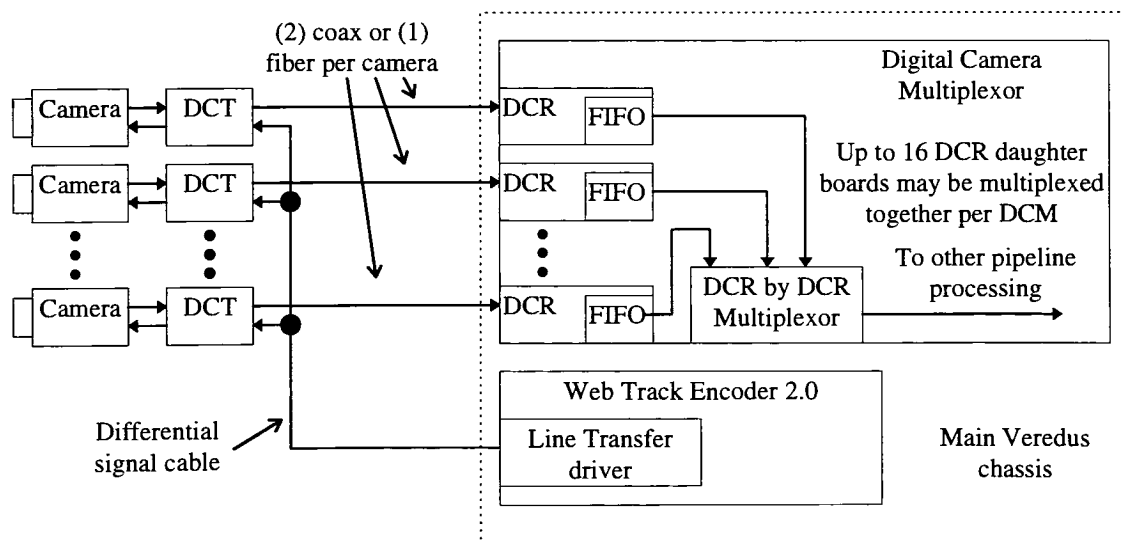
## **2.6. Camera to System Data Interface**

Digital data is transmitted from the cameras to the main Veredus chassis in systems presently under fabrication. Older systems drove differential analog data from the cameras on cabling up to 100 feet prior to being digitized in the main Veredus chassis. Power and control was provided to the cameras through the same cabling. Power voltage loss, power line noise addition, sensor signal degradation, and limitations on how far the cameras could be located from the main Veredus chassis drove the system design to digitize data in the cameras. Power supplies are now located near the cameras on the production line, allowing the main Veredus chassis to provide a differentially driven line transfer signal to the cameras, and receive back digital data.

To minimize the amount of cabling required, and maximize the distance data can be transmitted, the digital data is serialized. AMD TAXI chip sets are employed on the DCT to perform this function, while the Veredus designed camera employs a confidential chip set. For speeds below 125 mega-bits per second, and distances below 200 feet, a pair of coaxial cables can be employed to send the data from the DCT to the Digital Camera Receiver board in the main Veredus chassis. For distances up to 1 kilometer, and speeds up to 175 mega-bits per second, a fiber optic transmitter is installed on both the DCT and DCR. These bit rates allow use of 8-bit digital data at 10 million pixels per second, and 15 million pixels per second, respectfully. The Veredus designed camera allows 12-bit data to be transmitted at 20 million samples per second, through the use of a 300 mega-bit per second fiber optic connection.



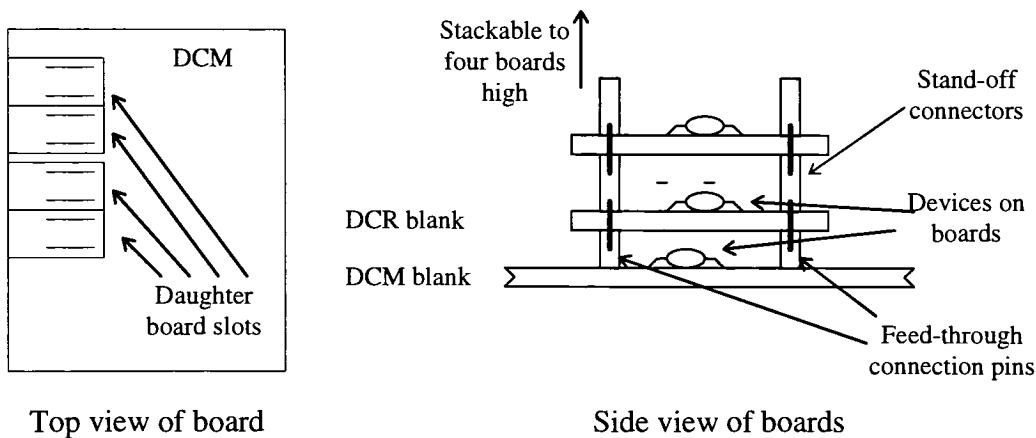
Each camera or DCT requires a separate DCR within the main Veredus chassis to receive its sent data, as shown in Figure 2-12. Different versions of the boards exist to handle the receipt of data from the Veredus designed camera, or the DCT/externally purchased camera combination. These boards are modularly placed on the Digital Camera Multiplexor board as daughter boards. This is accomplished by allowing four 2.5 by 4.0 inch daughter board slot locations on the DCM, and allowing the daughter boards to be stacked vertically on top of each other, as demonstrated in Figure 2-13. As a result, systems can be manufactured allowing from one to 16 cameras to be interfaced into a single Veredus DCM.



**Figure 2-12: Camera to System Modularity**

The multiplexing operation on the DCM relies on the existence of FIFOs on each DCR. The FIFO on each DCR is long enough to contain at least one camera scan line. The DCM can then read out one scan line from a particular DCR, while all other DCRs buffer

incoming pixels in their FIFOs. At completion of a DCR scan line, the DCM commences reading a scan line from the next DCR in the sequence. This allows multiple cameras to be placed across the width of a web, or even in multiple scanning geometries, and have the aggregate data treated as a single data stream by the rest of the Veredus processing pipeline.



**Figure 2-13: Digital Camera Multiplexor Daughter Board Connections**

## **2.7. Data Correction**

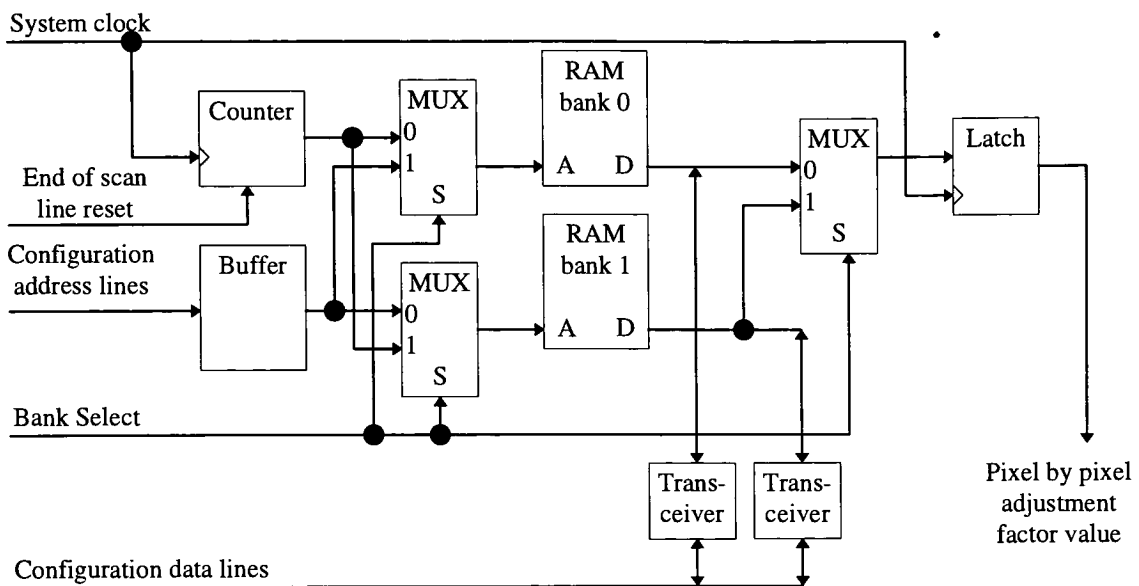
The DCM and A/D boards are capable of performing pixel-by-pixel offset and gain on 12-bit digital values. RAM tables are configured by software to provide an individual offset and gain for each pixel along the length of the scan line. Offset values are loaded into the appropriate tables on both boards to remove the effects of dark charge accumulation with the CCD sensor, and variations between offset values provided by multiple camera sources. Gain values are provided to account for unequal responsivity of the individual CCD array elements, whether from the arrays, the optical lenses, or from the illumination source.

Experimentation on production scanning lines demonstrated these values could not be static in nature, i.e., could not be loaded once at the start of a production run, and left unaltered. The signal from the illumination source on the production line and the scanned product consistently varies, making digitized values passed onto enhancement sections fluctuate as well. This consistently detracted from accurate detection of imperfections within the rest of the system. The addition of constant down-web pixel size line transfer clocks, and exposure control, further aggravated this situation. The employed solution is to dynamically change the pixel-by-pixel gain values as the system scans. This requires the use of double banked gain tables on the A/D and DCM boards, as shown in Figure 2-14.

To calculate the dynamic gain values, samples of offset adjusted (but not gain adjusted) pixel values are required. A separate running sum is calculated for each pixel across the

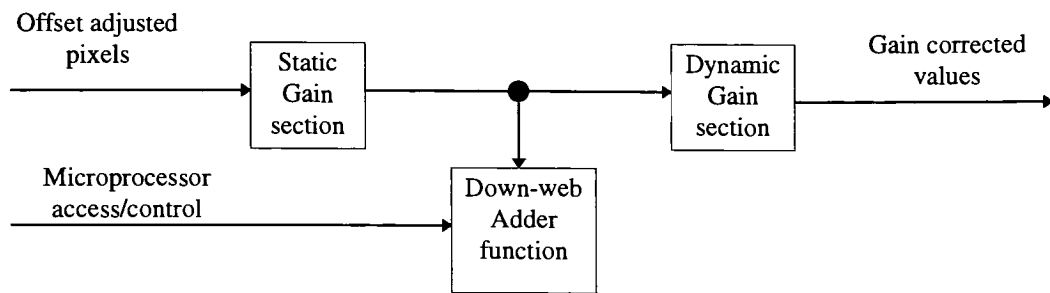
width of the scan line from these values. On the A/D board, this is accomplished by the Intelligent Filter Board capturing snapshots of gain adjusted values from the A/D output VME bus in memory. A microprocessor then removes the gain adjustment from the captured data by dividing by the current pixel-by-pixel gain values. These values are then used by the microprocessor to calculate the running sums. On the DCM, this process is augmented by a dedicated circuit (called a down-web adder) to capture data prior to the gain function. The down-web adder calculates the sums in real-time for a specified number of scan lines, and places the result in memory accessible to a microprocessor. The resultant sums are employed in a formula, along with the last gain values written to the pixel-by-pixel gain banks, to establish new gain values to employ. Effectively, the last gain values are employed by this low pass IIR filter to allow a history function, so that employed gain values do not fluctuate massively. These values are then written to an “off-line” pixel-by-pixel gain table on the A/D or DCM. Once the required length of the table has been filled, bits are set on the boards by software to command switching of the “on-line” and “off-line” gain tables at the completion of the present scan line.

Any line length from four to 32768 pixels is handled in the data correction circuitry on the A/D or DCM. The employed line length is software selectable on both boards, allowing software to only configure as much of the offset or gain RAM tables as is required for the present application.



**Figure 2-14: Double Banked RAM Gain Table**

Analysis with the A/D board on production lines also demonstrated a use for two gain functions. Refer to Figure 2-15. The first, static gain function would allow the structural components of the cameras and light sources to be removed. These values could be loaded at the start of a production run, and not require changing during the scanning operation, because the items it is intended to adjust do not change as the product moves. The output of this static gain section is then fed into a double banked, dynamic gain section. This allows dynamic adjustment to account for light level fluctuations as the product moves. Separation of these two sections allows the collection of data after static gain section, but before subtle, long running imperfections such as streaks, have been compensated out by the dynamic gain function. On the DCM, these multiple gain sections are provided, and work with the down-web adder circuit to allow detection of the subtle imperfections by microprocessors.

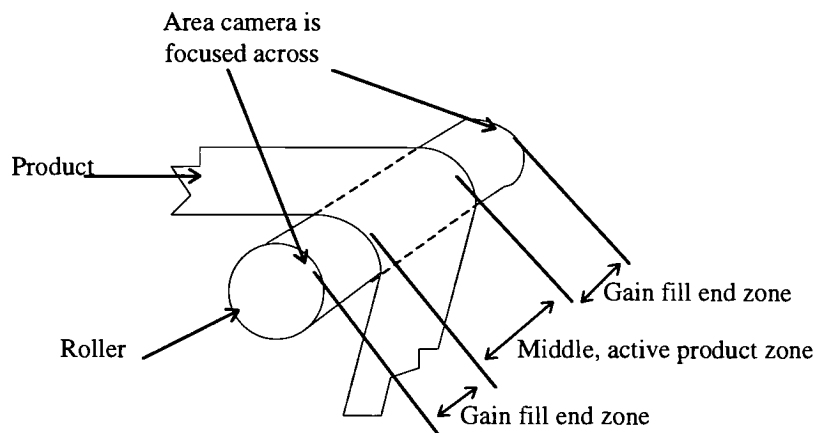


**Figure 2-15: Multiple Gain Sections Connections**

## **2.8. Edge Tracking**

In order to assure the entire width of a product under inspection is scanned, cameras are aligned to focus over an area wider than the width of the web. This allows detection of imperfections which exist close to the edge of the product. Any features that are detected beyond the edge of the product are not actual imperfections, and therefore must be removed from the data stream. In an ideal situation, circuitry or software could statically mask off the area of the web past the edge of the product by setting all pixel values in that area to a background level. Experimentation on production lines demonstrated a static mask is insufficient, due to the scanned web constantly weaving from side to side as it moves past the inspection station. This results in the need to dynamically establish the present edges of the product, calculate new mask values on the basis of the edges, and command circuitry via software to implement the new masks. This process is known as edge tracking.

In the A/D and IFB configuration, multiple pixel-by-pixel gain zones are established across the width of the scan line, as shown in Figure 2-16. A zone from the end of the scan line to around four pixels onto the product are established from both directions. The remaining zone is the area between these end zones. A specific “gain fill” value is employed in the gain tables written to the A/D by the IFB in these end zones. The middle zone corresponds to the area of active product, and dynamically calculated gain values are employed in this region. As pixel values are gained and passed out of the A/D board, the IFB captures snapshots of this data. The gain fill values allow the IFB algorithms to capture data along the edge of the product without the dynamic gain process having removed the actual transition from non-product to product. The running sums described above are again employed, and once complete, subjected to a one-dimensional cross-web edge detector FIR filter. This result is analyzed by the IFB microprocessor to establish the current edges of the product.



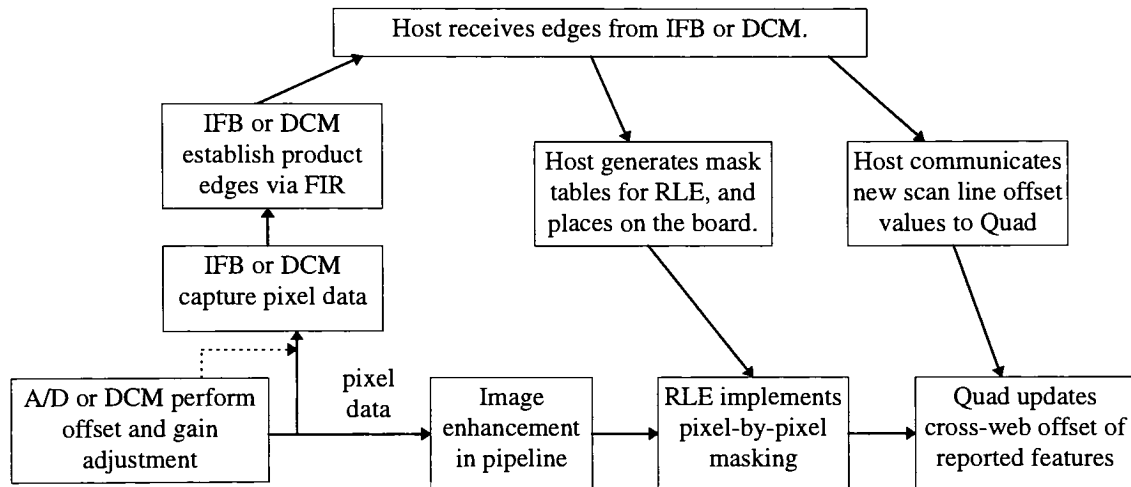
**Figure 2-16: A/D and IFB Edge Tracking Zones**

Enhancements implemented on the DCM allow the down-web adder circuit to directly capture data prior to dynamic gain adjustment. The resulting pixel-by-pixel running sum values are still subjected to a one-dimensional cross-web edge detector FIR filter. There is no longer a need to maintain gain fill zones at the end of the scan lines, since the concern over de-gaining the pixels near the edge of the product has been removed. The DCM microprocessor therefore establishes the present position of both product edges from the software filter output.

Detected product edges are reported to the Host microprocessor, as shown in Figure 2-17. The Host translates these values into pixel-by-pixel masking tables located on the RLE 1.6. Values located in these tables represent the function to perform on each pixel in the scan line, i.e., 0 passes the pixel value through, 1 sets the pixel phase value to background, and 2 deletes the pixel. The present Host algorithm supports only the concept of passing pixels, or setting the pixel to a background level. Because there is a



need to dynamically change these values during run-time, double banked RAM tables are employed similar to the dynamic gain tables of the A/D or DCM. The Host therefore writes a new table down to the off-line RLE 1.6 masking RAM bank, and instructs the board to make the table active by setting a bit in the board's register space. At the end of the present scan line, the RAM tables swap, and the new product masks are employed.



**Figure 2-17: Implemented Edge Tracking Data Flow**

The updated edge locations are also employed in the connected components algorithm running on the Quad Board to accurately locate and report the cross-web position of any imperfections. This updated information is handed from the Host to the each Quad processor through a complicated software handshaking mechanism. Co-ordination of providing new masking table values to the RLE 1.6 and cross-web offset values to the Quad Board is the responsibility of the Host processor. Results have shown this consumes a significant amount of Host processing power, and limits the responsitivity of the edge tracking function.

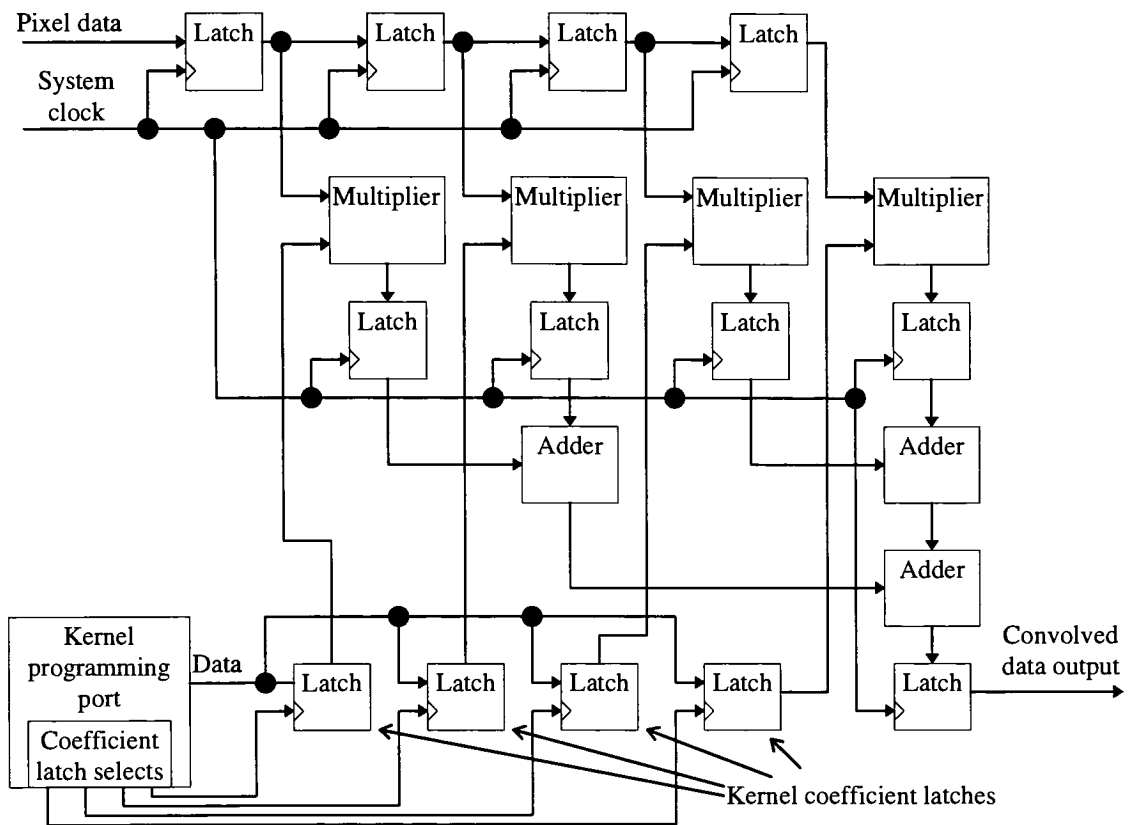
To alleviate the need to include the Host in the edge tracking cycle, circuitry has been provided on the DCM to allow masking of pixel information prior to enhancement and analysis by the rest of the Veredus pipeline. This masking circuitry provides a pixel deletion capability, up to a four pixel look-ahead feature, a repeat the last pixel option, and a pass the pixel through choice. The envisioned controlling algorithm effectively deletes all pixels in the scan line up to the first edge of the product on each scan line. The algorithm passes through all active product pixels on the scan line until the second edge, and then repeats the last pixel prior to the second edge until the scan line is complete. Implementation of this concept reduces the edge tracking function to be local on the DCM, and removes the need to adjust RLE 1.6 masks, or Quad Board cross-web offsets. Plans must still be established to incorporate this feature in system software.

Implementation of edge tracking becomes more complicated when the ability to include multiple cameras in the same view of the web, or to include multiple web views within the same pipeline, is required. With multiple cameras in the same web view, the end pixels of the cameras are purposely overlapped by several pixels. This assures complete scanning coverage of the web. It also allows any edge effects created by image enhancement of the degraded signal observed at end of a sensor to occur in an area covered by an adjacent sensor. However, the same imperfection cannot be reported multiple times, and false edge effects should never be reported. The overlap must therefore be removed prior to reporting of any imperfections. Multiple web views in the same pipeline simply requires duplication of the edge tracking function within the same

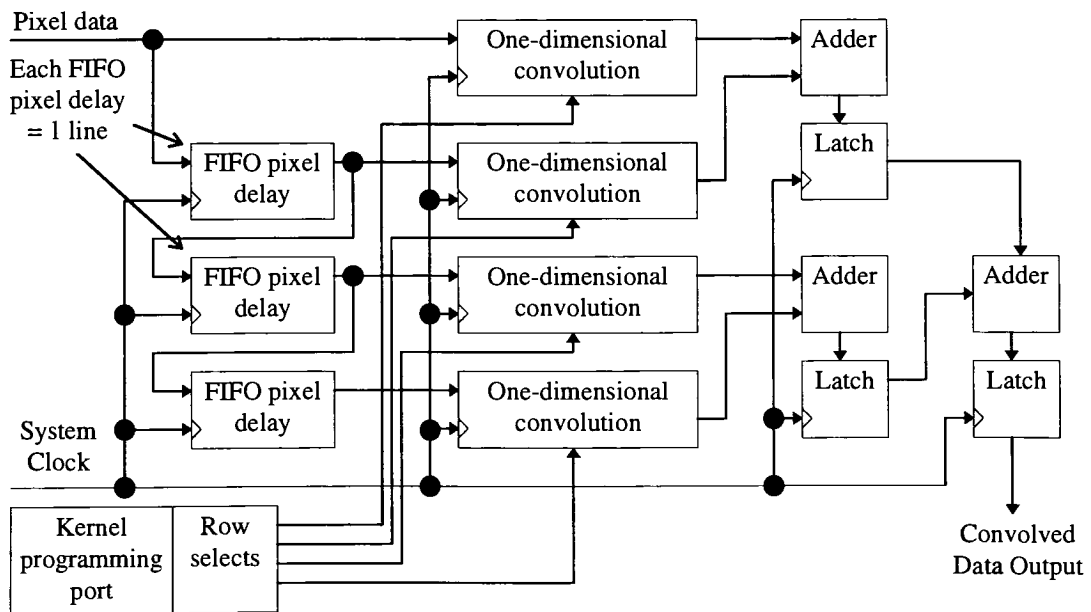
pipeline. The implemented hardware supports the need to delete overscanned pixels at camera boundaries, and to isolate independent web views within the pipeline from each other. Software must configure the boards to allow appropriate use of these functions.

## **2.9. Enhancement**

Image enhancement within the Veredus relies on the use of two-dimensional FIR convolution. This mathematical function is accomplished in the one-dimensional continual flow-through environment by pipelining multipliers and adders, as shown in Figure 2-18. To transition to the two-dimensional world, the pixel data stream must be buffered to allow the presentation of the same cross-web pixel in the image to the FIR process from multiple scan lines simultaneously. The use of FIFO memories, and then the summation of the one-dimensional FIR results from each scan line, allow the implementation of a continuous flow-through two-dimensional FIR convolution, as presented in Figure 2-19. The Convolver 1.0 board implements a 5 x 8 convolution by cascading five one-dimensional FIR chips, intermixed with FIFO delay modules, in this manner. On the Convolver 3.0, a single device performs the multiplications and additions required for an 8 x 8 convolution, with external FIFO memories providing scan line delays. Kernel coefficients are configurable by system software on both boards.



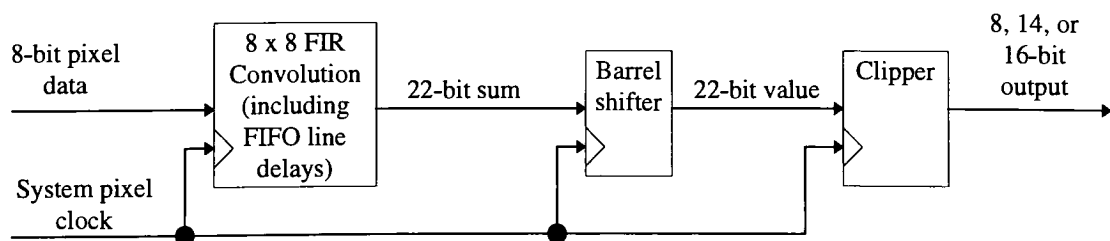
**Figure 2-18: Implementation of 4-pixel Wide One-Dimensional Pipelined Convolution**



**Figure 2-19: Implementation of 4-line Two-Dimensional Pipelined Convolution**

The employed 8 x 8 convolution on the Convolver 3.0 allows 8-bit pixel data, and 8-bit two's complement coefficients. This results in a potential 22-bit sum for each calculated FIR sum, as shown in Figure 2-20. The same chip provides a right-shift barrel shifter, allowing reduction of the passed convolution sum value as requested by software. This value is then passed through a clipping circuit, forcing enhanced pixel values to be held or a 8, 14, or 16-bit value. Typically the 14 or 16-bit values are employed as outputs of the convolution section.

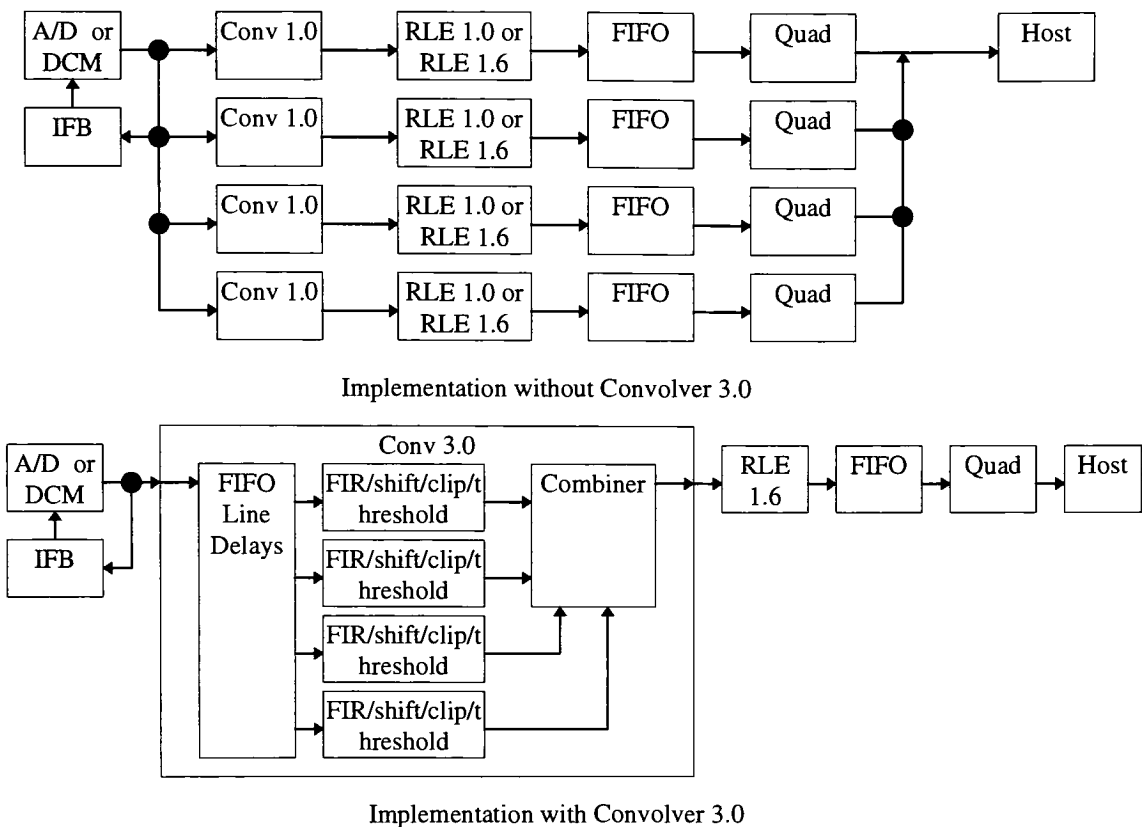
In contrast, the Convolver 1.0 employs 8-bit pixel data, but only 4-bit coefficients. With the 5 x 8 kernel size, the convolution sums can take on values from -81,600 to +71,400. At board implementation, and in subsequent use, it was envisioned all coefficient values would never be set to the maximum (+7), or minimum (-8), value in the same application. In practice, AC kernels are almost always used, allowing the design to be successfully implemented with the assumption output values never exceed a 16-bit value (-32768 to +32767). The lower 16-bits of the convolution output are therefore passed onto processes following the FIR function, without passing through a barrel shifter or a clipper.



**Figure 2-20: Convolution, Barrel Shifter, and Clipper Data Flow**

Analysis with the Convolver 1.0 board in production settings demonstrated a need for multiple kernels to be used on the offset and gain compensated pixel data provided from the A/D board. For instance, one kernel would be employed for detecting spot type imperfections, and a different one would be employed for detection of edges. To provide this functionality, the same A/D output data is sent to multiple Convolver 1.0 boards, with a full compliment of RLE, FIFO, and Quad boards following the Convolver as well. The A/D output appeared to the Host as having multiple reporting pipelines, one for each convolution kernel employed.

To accomplish parallel convolutions in a more efficient manner, the Convolver 3.0 board provides four parallel FIR chips, barrel shifters, and clippers. See Figure 2-21. The same FIFO line delays feed all four FIR devices. The output of the clippers are passed through separate threshold look-up tables for each enhancement path on the Convolver 3.0, and the results are combined together in a “combiner” circuit. The output of the combiner circuit sends a software configurable 4 or 8-bit “combined phase” value per pixel onto data reduction on the RLE 1.6. This results in a significant reduction of the number of circuit boards required in the system.



**Figure 2-21: Parallel Convolution Implementations**

To allow analysis on the unenhanced pixel data by boards following the Convolver 3.0, a data path with appropriate pipeline delay is provided around the FIR convolutions. This unenhanced pixel data is then emitted on the address lines of the output VME bus in parallel with combined phase data on the VME bus data lines. Specialized circuit boards have been designed for particular applications, which use a change in the combined phase data as a trigger to restart an analysis of the unenhanced gray scale data. For example, if the combiner phase output has a value of four for a series of pixels, and changes to a value of three, the analysis on the gray scale data will restart. The results of the analysis while the phase value was four will be logged, and passed on as needed. Calculations

that have been performed include the peak gray scale value detected, and an average of the gray scale value along the phase run. In this circumstance, these value are then employed in the classification of detected imperfections.

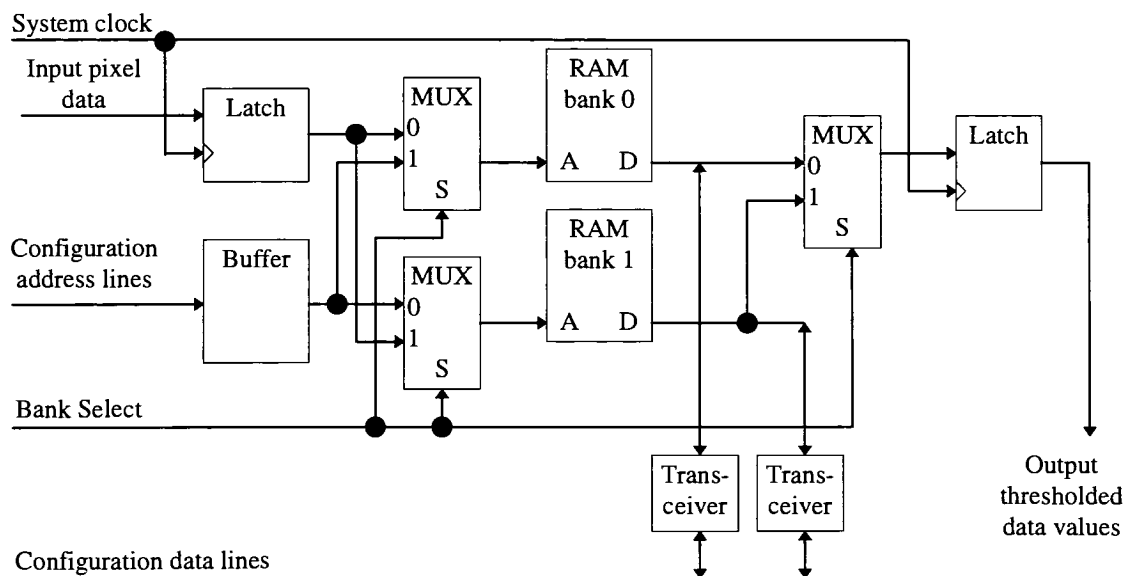
## **2.10. Thresholding**

The RLE 1.0 implemented a single banked, generic type LUT to provide convolved output to phase value thresholding. A 16K x 2 RAM table is provided, which requires ignoring the lowest 2 bits of the 16-bit input convolved values. This allows threshold values from -32,768 to +32,767 in steps of four. For example, convolved values of 0 through 3 map to the same RAM location, 4 through 7 map to the same location, etc. Due to the two bit phase value, phases of 0, 1, 2, and 3 can be placed in the table. Typically software configures the tables with a value of 0 to represent a background phase value, 1 to show a deflection to the positive side of background, and 2 to indicate the threshold below the background signal has been crossed.

Production line experimentation with the RLE 1.0 identified several areas of improvement. First, each time the threshold value is changed, the entire image processing pipeline has to be stopped, since the tables can only be updated reliably when the board is not clocking data. When attempting to install and certify systems, thresholds are changed many times to establish the proper threshold values during each product roll. Each time they are changed, the process of stopping and starting the pipeline requires up to a minute for the system to reload all the configuration tasks, and then switch back to the acquisition mode. During this time, product passes by which is not inspected.



Additionally, some applications have proved allowing the Veredus to dynamically calculate and change the threshold values, on the basis of statistics the system software gathers, enhances accurate imperfection detection. Double banking the threshold tables allows the system to continue scanning as a different threshold table is placed in the RLE 1.6, as shown in Figure 2-22. The new table is enabled by software setting an appropriate swap bit on the board. At the end of the present scan line, the RLE 1.6 swaps the active threshold bank off-line, and places the off-line bank on-line.



**Figure 2-22: Double Banked Threshold Table**

Production line experimentation also showed additional phases are occasionally useful. Multiple phase values on each side of the background signal level allow a determination of how far away from background a particular pixel is after it has been thresholded. To allow for this, and obtain devices which manufacturers felt would be supported for a

reasonable amount of time, 64K x 4 RAM chips were chosen to implement the RLE 1.6 LUTs. This allowed threshold steps to become increments of 1, from -32,768 to +32,767.

Twenty MHz operation within the Veredus pipeline mandates two pixels must be passed per VME bus data transfer cycle. In addition, only eight bits are allowed per pixel transferred. The transferring of data on address lines does allow 16 bits per pixel, but two different forms of data are transferred in this case, still forcing a requirement for eight bits per pixel. The output of the Convolver 3.0 convolution clipper results in 14 or 16 bits per pixel. Reducing this value down to only eight bits per pixel for transfer to a following threshold board produces an unacceptable granularity in the system sensitivity. The situation is further aggravated by placing four parallel FIR enhancement paths on the Convolver 3.0. Threshold tables are therefore placed on the Convolver 3.0 to allow reasonable system operation. To allow for this feature, the RLE 1.6 has the ability to bypass the threshold function when receiving data from the Convolver 3.0.

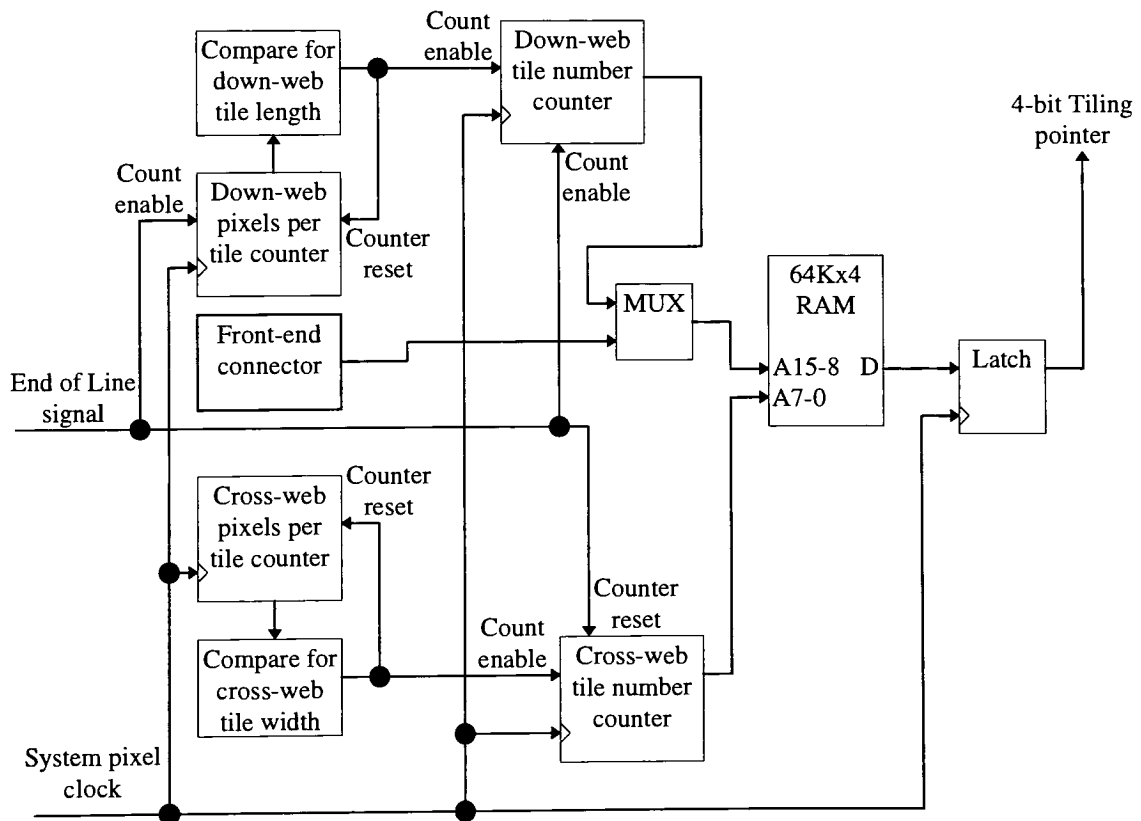
The combination of multiple cameras through a single pipeline serves as a cost savings in the Veredus. Because each camera has a slightly different responsitivity, selecting a universal threshold value for all multiplexed cameras results in some cameras producing more false hits than others, and some missing imperfections. By providing independent thresholds for each camera, this problem is alleviated. In addition, sheet metal products that have been cut into sheets display the nature of a continuous web process, with areas of “blank” web which do not need to be scanned. By forcing a background phase value out of the threshold LUT during the time when product does not exist under the camera,

and employing an actual threshold table when product does exist, the Veredus can be applied to scanning sheet type products. Both these issues are addressed on the Convolver 3.0 by the implementation of a tiling function.

To allow the tiling function to point to multiple logical threshold tables on the Convolver 3.0, 256K x 4 RAM chips are used to implement the LUT. The LUT is treated as a single, 64K x 4 threshold LUT when tiling is not required, ignoring the upper three quarters of the device. When tiling is desired, the physical LUT is viewed as 16 separate 16K x 4 logical threshold LUTs. A 4-bit tiling pointer is routed to the upper address lines of the threshold LUT RAM devices. Different threshold functions are therefore loaded into each employed logical threshold bank, and the tiling pointer selects which logical threshold should be used on the basis of where the present pixel is within the scan line or image.

The tiling pointer is generated by accessing a 64K x 4 tiling RAM bank. Refer to Figure 2-23. The lowest eight address lines of the RAM are used to access tiles in the cross-web direction. At the start of the scan line, one counter acts as a pointer to the RAM table, with a value of zero on the RAM address lines. A different counter then counts the number of pixels, which have passed in the present tile. When this number matches a value configured by software, the second counter resets, and the first counter is incremented by one. This process continues across the width of the web, at which point the first counter is again set to zero to start the next scan line. The accessed contents of the tiling RAM provide the 4-bit tiling pointer which is employed in the threshold tables.

The upper eight address lines are driven either from a dual counter configuration as described for the cross-web case, or by a 4-bit connector located on the front of the Convolver 3.0. The down-web dual counter configuration allows the web to be divided into a two-dimensional grid. This feature is typically configured to only employ one down-web pointer value, which reduces the concept to cross-web tiling only. The front end connector allows the receipt of a signal from an external sensor to switch between cross-web tile groups (i.e., groups of 256 tiles in the cross-web direction), which works well in a sheet type application.

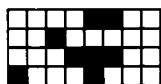


**Figure 2-23: Tiling Function Implementation**

Configuration of the tiling section is non-trivial. But the effect can significantly reduce software timing requirements for control of the thresholding function. Earlier sheet type system installations required the use of interrupts to switch between active scanning areas, and non-product areas. The interrupt handler had to change the pipeline configuration within a short amount of time to avoid scanning what should not be scanned, or vice-versa. Connecting the sheet sensor to the front of the Convolver 3.0 allows dedicated hardware to perform the scan/don't scan function, and significantly relaxes timing constraints on software processes.

## 2.11. Data Reduction

The RLE 1.0 board implemented data reduction through the concept of Run Length Encoding. A simple mechanism is used on this board: counting the number of sequential pixels at the same phase. See Figure 2-24. When a phase change is detected, (i.e., changes from phase 0 to 1, etc.), the count of the number of sequential pixels at that phase is combined into a single 16-bit word with the phase value. Because two bit phases are supported, count values can range from 1 to 16383. No allowance exists in the implementation for the end of a scan line, so a run length encode frequently includes information from multiple scan lines. This is known as the RLE-I encoding format.



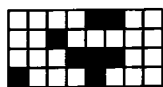
With white = phase 0, and black = phase 1, RLE-I style codes from this 8 pixel wide and 4 line long image report:

Phase 0, length 4	Phase 1, length 2	Phase 0 length 4
Phase 1, length 1	Phase 0, length 8	
Phase 1, length 3	Phase 0, length 2	
Phase 1, length 1	Phase 0, length 3	Phase 1 length 1    Phase 0, length 3

**Figure 2-24: RLE-I Encoding Format**

Experimentation showed any process receiving the reduced data had to re-establish a two-dimensional view of the web under scan prior to analysis of any detected features. Connectivity algorithms were implemented in software which do this with the RLE-I style reduction scheme in an efficient manner. Transitioning the connectivity process to use hardware accelerators was determined to be more straightforward by reporting phase changes in a two-dimensional space.

The RLE-II encoding format allows for multiple words of data in each run length encode. The number of bits allowed for the phase information is expanded to four. Reporting the end of a sequential phase run provides a 15-bit cross-web and 28-bit down-web value. At the start of a product roll, both the cross-web and down-web counters are cleared. As each pixel is received in the encoder, the cross-web count is incremented by 1. At the software configured end of scan line, the cross-web counter is zeroed, and the down-web count is incremented by 1. Each time a phase change is detected, the value of the cross-web and down-web counters are reported with the phase which has just ended. In addition, the end of any scan line that has had a phase change on it is also reported, to provide a clear understanding the end of line has been reached. The information reported to processes following the run length encoder is therefore in a two-dimensional co-ordinate system, as shown in Figure 2-25. Each reported phase change indicates an absolute position within the scanned product roll of where the change occurred. This assists the application of hardware accelerators to the connectivity process, since the two-dimensional nature of the web is preserved in the reported run length encode information.



With white = phase 0, and black = phase 1, RLE-II style codes from this 8 pixel wide and 4 line long image report:  
 Note X and Y values are ending pixel number locations

Phase 0, X=4, Y=1	Phase 1, X=6, Y=1	Phase 0, X=8, Y=1
Phase 0, X=2, Y=2	Phase 1, X=3, Y=2	Phase 0, X=8, Y=2
Phase 0, X=3, Y=3	Phase 1, X=6, Y=3	Phase 0, X=8, Y=3
Phase 1, X=1, Y=4	Phase 0, X=4, Y=4	Phase 1, X=5, Y=4    Phase 0, X=8, Y=4

**Figure 2-25: RLE-II Encoding Format**

With the one word long run length encode requirement removed by implementing the RLE-II multiple word encodes, the addition of more information about a phase run is

straightforward. For one application, unenhanced gray scale data has been provided from the Convolver 3.0 board in parallel with the reported phase data. Circuitry placed along side of the RLE-II encoder is commanded to perform calculations on this gray scale data. Per run length encode, the maximum gray scale value encountered is calculated. A sum of the gray scale values over each phase run is also calculated. These calculated values are grouped together into an additional word of information, and appended to each RLE-II run length encode. The connectivity process receives these additional words, and incorporates them into a feature by feature maximum gray scale value, and average gray scale value. These additional pieces of information are employed in the classification process to assist in accurate imperfection identification.

## **2.12. Connected Components and Classification**

The process known as connectivity and measurement is performed on the Quad II board. A software algorithm executing on the four parallel Motorola 68010 microprocessors reads run length encode values from the FIFO board, and performs a multiphase connected components analysis. Measurements that are calculated on each resultant feature are width, length, and area of each phase. The software algorithm attaches a classification field to the vector of data for the feature representing the source of the imperfection, i.e., which logical view within the pipeline, and which pipeline number. Feature clustering algorithms and rudimentary feature removal options may be enabled in the connectivity and measurement programs by higher level (Host) configuration tasks. Resultant feature vectors are reported to the Host microprocessor for additional classification, including repeats analysis.



The implemented algorithm places several requirements on the hardware architecture to support this process, and vice versa. Employed run length encoding processes do not lend themselves to striping the web in a vertical direction, i.e., allowing the first processor to analyze the first quarter of each scan line, the second to receive the second quarter, etc. Further, interprocessor communications allow writing of information into a common static RAM bank, but do not have sufficient bandwidth to share information about a scan line between processors in real time. Experimentation showed the only effective use of the quad processors was to stripe the data in the horizontal direction. Each processor reads a certain number of scan lines from the FIFO board into its local memory, and notifies the next processor through common memory it is time to switch readers. Once this is performed, the first processor performs the connectivity and measurement process, and reports calculated features to the Host.

The process of dividing the web into horizontal stripes generates false imperfection termination points. When a processor completes the acquisition of run length encode data, and passes FIFO board control to the next processor, it only reports the last phase and number of pixels at that phase which were left over at the end of the scan line. The next processor starts with no open features, since there is no way to hand this information between processors in a timely manner. To complete all the processing to understand if features should exist across the horizontal stripe boundary, the first processor would have to complete all of the connectivity and measurement for the data it has locally stored. If this could be done in a timely manner by the first processor, there would be no need to

employ multiple processors in a striping manner, because all required processing could be completed by only one processor. Hence, the effect of horizontally striping the data allows a pipelining of the connectivity and measurement process. It also breaks any imperfections that cross the horizontal stripe boundary into multiple features, which are reported from multiple processors.

To increase the rate at which features can be handled by the pipeline, and reduce the number of features that are divided by horizontal striping, a Connectivity and Measurement Board was designed. This board captures RLE-II codes, and breaks each portion of the code into a separate data stream. A hardware state machine employs the data streams to calculate all boundary data items needed to establish the connectivity of each feature, along with some of the rudimentary measurements. An algorithm residing on a single controlling Texas Instruments Digital Signal Processor accesses appropriate memory mapped locations to complete the measurements, and will report the resultant features to the Host. Analysis on this board has demonstrated a tenfold increase in the number of features which can be measured per lot time compared to the present Quad 68010 implementation, without a need for horizontal striping. Efforts to complete software integration of this board design, for both the local microprocessor and Host microprocessor, must still be concluded.

### **2.13. Interprocessor Communication**

Several modes of interprocessor communication are employed in the Veredus. With the first, the Host processor provides information to sub-ordinate processors on the Quad, IFB, or DCM by stopping any processor operations underway on the slave board. Information is written to the slave microprocessor's local memory by the Host. The slave processor is then placed back in an operating mode by the Host when the data transaction is complete. This mode works well to load a new programs into a slave microprocessor's memory space. Unfortunately, it also requires the slave microprocessor to stop executing if communication information is transferred between the Host and the slave by this mechanism. This results in a loss of potential parallelism of the processors. If there is no way to recover execution at the last instruction executed before the Host stops the slave microprocessor, it also requires a slave microprocessor to place itself in an infinite loop after it has notified the Host it has information to provide.

To enhance the ability of the slave microprocessor to keep executing after data is made available to the Host, the IFB provides a FIFO buffer memory which the local microprocessor may write to. Once the IFB writes this data, it is free to continue processing. Additional processing time is provided in this manner, but no FIFO exists coming from the Host to the IFB slave processor. The slave must still be halted for the Host to send information to it. The Connectivity and Measurement Board provides two FIFO buffers, one in each direction, to alleviate this problem.

On the DCM, a dual ported memory is employed. The chosen device provides semaphore locations for both Host and slave microprocessors to use. This assures, through software protocols, that the same memory area is not being accessed by both processors simultaneously. Both processors can write communications information to the dual ported memory, and proceed with other processing.

All Veredus boards with microprocessors on them can interrupt the Host. This allows the slave boards to notify the Host when they have placed something in a memory buffer which the Host must know about. In addition, the DCM provides the ability for the Host to interrupt the slave microprocessor.

On the Quad II board, communications between the four parallel processors is provided. Single ported static RAM is accessed through a prioritized arbitration scheme when the processors access an appropriate memory range in their address map. This scheme prevents bus contentions, but does not offer any fairness protocol. Further, there is no means for processors to signal each other that information exists in the common memory for access. The algorithms operating on the Quad-II board do not require many interprocessor communications, but this small amount of interprocessor communications is partially driven by the restrictions of the board design.

## **2.14. Event Tracking**

Events occurring on the production line have bearing on the operations of the Veredus pipeline. The beginning of a product roll, the end of a product roll, an area where known waste product is being generated, etc., are all reported to the Veredus through digital inputs. The Host microprocessor receives these inputs and performs appropriate action to account for them. Production line experiences demonstrated forcing this information down into the image processing pipeline could enhance the system's response to these external events. Additionally, some of the processing performed by the Host processor could be off-loaded by providing additional information about the production line status to slave microprocessors in conjunction with image information.

The Data Organizer concept allows external event, internal event, and web position information to be transmitted down the length of the Veredus pipeline. External event and web track encoder information is obtained at the production line, and injected into the data stream at the camera, when employing high end Veredus designed cameras. These event fields are included once per each scan line obtained. This data is transmitted to the main Veredus system over the fiber optic cables, and transferred through the Digital Camera Receiver board to the Digital Camera Multiplexor board. The DCM adds an internal event field to report items occurring within the Veredus pipeline which may be of value to future boards in the pipeline. The event field then propagates through the Convolver 3.0 board, and arrives in the RLE 1.6. The web track encoder information is then substituted for the absolute down-web count value in reported run length encodes. External and internal event information is included as special event run length encode

values reported to processes following the RLE 1.6. This allows the connectivity and measurement process to have an understanding of web position, and activities of both the outside world and prior activities of the pipeline.

Full use of the Data Organizer concept has not yet been completed in Veredus system software. When complete, its use will allow slave processors to calculate perimeters in either constant frequency or constant pixel mode. Additional classification by slave processors will be possible due to knowledge of down-web position, including repeats analysis. Knowledge of external and internal event information allows an understanding of sudden changes in feature data loads, and the termination of open features in connectivity and measurement when conditions warrant.

## **2.15. Diagnostics**

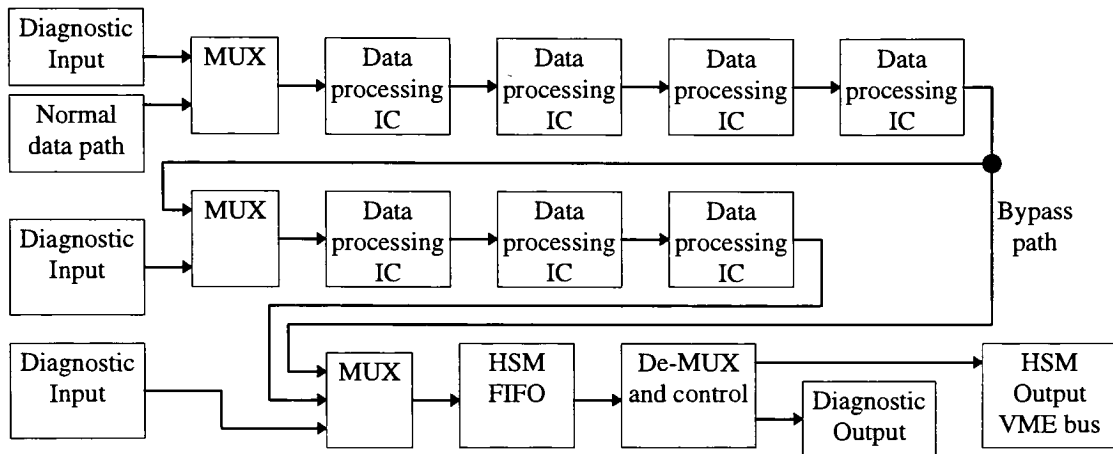
Early versions of pipeline boards included few diagnostic options. The RLE 1.0 board provided a bypass path to route all data passed into the board to the output of the board. The A/D board provided a “gray scale” register which could be enabled in lieu of the Analog-to-Digital converter output. It was therefore possible to implement a diagnostic data pattern by loading various gain values in the A/D board gain compensation memory, and adjusting the A/D board gray scale values as required. The Intelligent Filter Board allowed the ability to snapshot data off from the VME bus, and analyze it with a microprocessor. These combinations allowed some form of onboard or system level diagnostics. Most board level diagnostic functions required placing pipeline boards in a specialized test chassis, and employing special diagnostic boards to pitch data patterns,

and to receive the resultant output data. If a failure occurred in a device on the board under test, technicians could not isolate the problem area without tracing data through the entire board processing path.

Later board designs improved upon the functional diagnostic capabilities located on the boards. Because FIFOs are required at the output of all boards with High Speed Master VME interfaces, a diagnostic output feature is included in the designs. When enabled, data is buffered in the FIFOs, rather than being sent to the output VME bus. The Host microprocessor can then read the data out of the FIFOs, to verify data patterns which have passed through the boards. Further, Host accessible diagnostic input ports are provided that allow the injection of data into the boards to be tested. Dependent upon the specific design, multiple input ports are provided, to allow better determination of where problem areas are on tested boards. Diagnostic input data usually feeds into a board at a Host accessible data rate, which is significantly less than the full 20 MHz data rate the boards operate at. In some implementations, a diagnostic input port provides a means to pump data through the board at full speed by feeding data into the input of the High Speed Slave FIFOs.

A full gamut of bypass functions have grown to allow multiple detours around processing functions into the input of the HSM FIFOs. Combining the bypass and diagnostic input/output features together allows the testing of an initial short data path on boards to verify a base level functionality of four or five devices in the data path. See Figure 2-26. The options are then repetitively changed to allow data to flow-through an additional four

or five devices, until the entire functionality of the board is tested. The bypass functions are also employed during general system operation, to avoid configuring sections of circuitry that are not being employed.



**Figure 2-26: Diagnostic Input, Diagnostic Output, and Bypass Implementations**

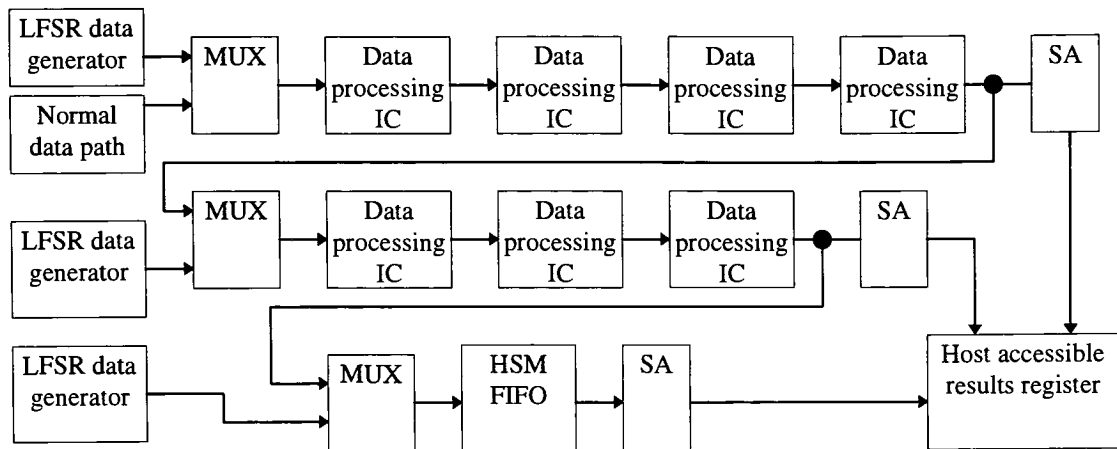
The diagnostic input, diagnostic output, and bypass functions are employed in system level diagnostics. Most test bench level software can be directly applied to the boards when installed in the system. Further, the diagnostic type patterns configured on the A/D board are also applicable to the DCM. With the existence of diagnostic output ports and bypass options along the length of the pipeline, a detected failure from an overall pipe functionality test can be diagnosed to the particular board which is failing. Board functions can be successively circumvented until a proper result is obtained, resulting in a knowledge of which boards are performing properly, and which are not.

To assist in verification of Digital Camera Transmitter and Digital Camera Receiver boards, a diagnostic pattern can be enabled on each board. This incrementing ramp



function (starts at 0, and increments by one on each successive pixel until 255, then repeats) allows quick verification all bits are toggling, and line synchronization exists. The offset table in the DCM can be configured to adjust this pattern to provide the same values for each every pixel, simulating data coming from the DCM gray scale register. Therefore, any system level diagnostic that exists using the A/D or DCM gray scale register to verify system level operation can also verify the DCT, DCR, multiplexing circuitry on the DCM, and any connections between these boards.

The Convolver 3.0 board provides hooks for the use of Built-In Self Test (BIST), as shown in Figure 2-27. Space is allocated in programmable logic to generate pseudo-random data sequences employing Linear Feedback Shift Registers (LFSR), and to capture circuitry results with Signature Analyzers (SA). Circuitry is divided into groups of four or five data handling devices, into which the LFSR data is injected. As necessary, these LFSR sequences have the capability to configure the data handling devices as well, such as RAM LUT tables. The output of each data handling device is analyzed by a SA, and compared to an expected value contained in the programmable logic. The result of each SA compare is transmitted in a single bit to a central reporting register on the Convolver 3.0. A test of nearly all functions on the Convolver 3.0 could be conducted within two seconds employing the BIST function. This powerful diagnostic capability is not currently exercised on the Convolver 3.0, since establishment of appropriate LFSR sequences, SA sequences, and comparison values within the programmable logic are not yet complete.



**Figure 2-27: BIST Conceptual Implementation**

Recent microprocessor based designs have predominately employed Texas Instruments Digital Signal Processor devices. Specifically, TMS320C30 and TMS320C31 have been employed. A significant reason for the choice of these devices is the existence of an emulator connection. By allocating a 12-pin header on an implemented board, an emulation tool connected to a PC compatible allows an extensive software debug aid. Source code may be downloaded by means of this port, and executed in single or multiple steps. Memory and register locations can be viewed and changed through the port. No jumpers or switches must be changed on the implemented board designs to facilitate the installation of the emulation tool. This has resulted in significantly reduced software and hardware debug and verification efforts in all designs employing these parts, as compared to designs which have employed devices which do not offer similar diagnostic tools.

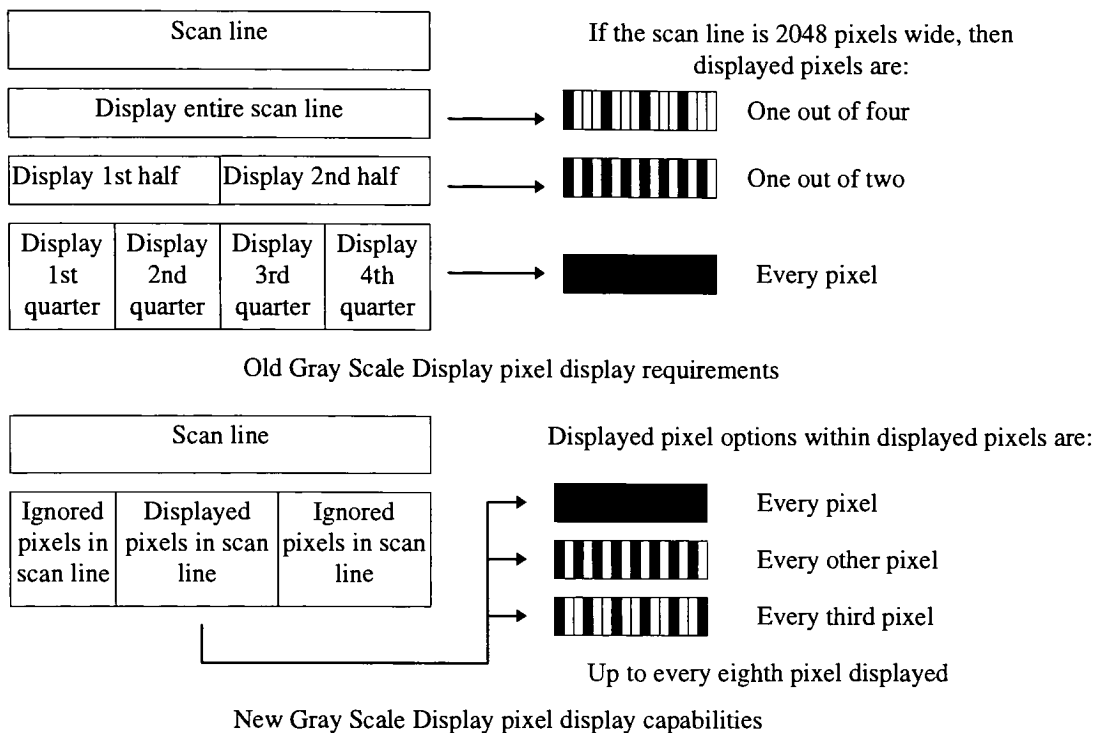
## **2.16. Gray Scale Display**

Multiple generations of Gray Scale Display Boards exist for use in the Veredus. The first board can capture continuous flow-through pixels, and display them as frames to the user on an RS-170 compatible display. This allows 512 x 512 frame sizes. Additionally, frame sampling exists, allowing a frame to be captured and displayed, while other frames are ignored by the capture circuitry. This works on a power of two increment, so that one frame may be viewed while one is ignored, one viewed as three are ignored, etc. This provides the user additional time to view displayed frames of data, allowing a visual analysis. System software provides the user a capability to stop the acquisition of data, and store a current frame for later retrieval and analysis. The frame sampling therefore allows time to determine if a useful image is on the display, and stop acquisition of a new frame.

On site experimentation with the first generation Gray Scale Display Board defined several areas where improvements were desired. Internally generated addresses showed where data should be read from the internal image display memory. The writing of flow-through data into the display board's memory was controlled by addresses coming from the VME bus. Data could only be sampled across the scan line by ignoring bits in the address range because of barrel shifting. The result was division of what could be displayed to halves of the web, quarters of the web, etc., as shown in Figure 2-28. As a wider view of the web was taken, pixels were sampled to create a view of the web. For instance, a 2048 scan line width displayed every pixel if a quarter of the web was chosen, since the display is 512 pixels wide. Choosing to display half the web forced every other

pixel to be dropped when data was written to the display board memory, since 1024 pixels are being mapped into 512 display locations. Additionally, addresses are not always provided from Veredus designed image processing boards, due to a need to use those lines for additional data transmission. When combined with non-trivial hardware reconfiguration of the board to view data other than pixel data provided by the A/D or DCM, the number of locations where the board can be used is limited.

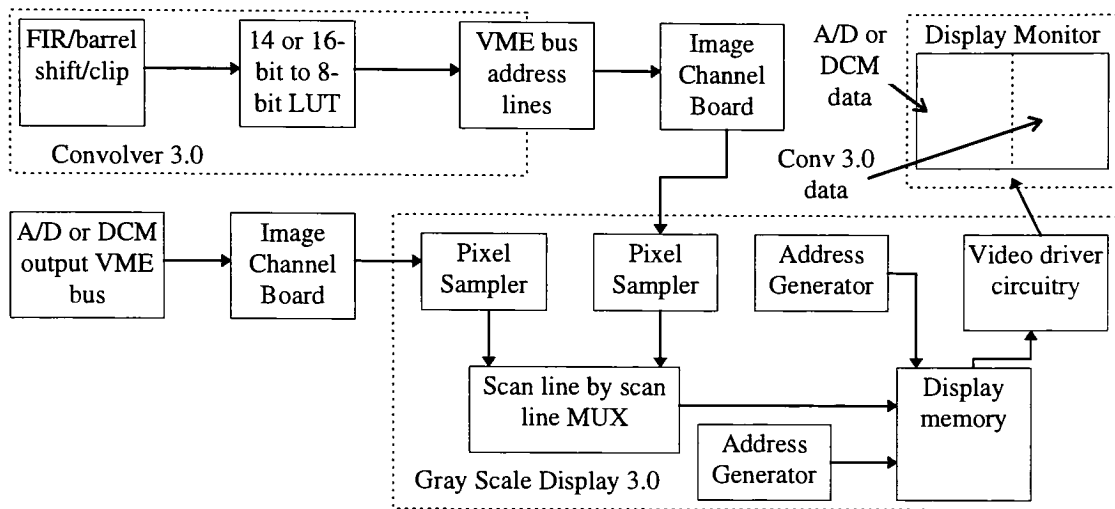
A newer generation of Gray Scale Display boards resolve these shortcomings, and provide other enhancements. Independent addressing for both writing incoming pixel data to memory, and reading pixel data for display, is provided. Sampling circuitry is therefore possible to select where on a scan line to start, and indicate how many pixels are desired from the scan line once the sampling starts. In the sampling range, software can configure if all pixels should be captured, if every other one should be captured, if every third, etc. VME bus data is acquired through a slave Image Channel board, and ribbon cables, allowing easy hardware reconfiguration as different data is required. Data from up to four data paths (independent Image Channel boards) may be captured and displayed at the same time, in separate vertical stripes. Frame sampling has been maintained, but no longer requires the presented frame rate to be based on a power of two value; the number of ignored frames can be any integer value from one to 15. Display capabilities for the video driver have increased to 1024 x 1024 pixels at 16 million colors, but the usual configuration is 1024 x 768 with 256 gray scale values presented.



**Figure 2-28: Example Modes of Sampling Display Data from a Scan Line**

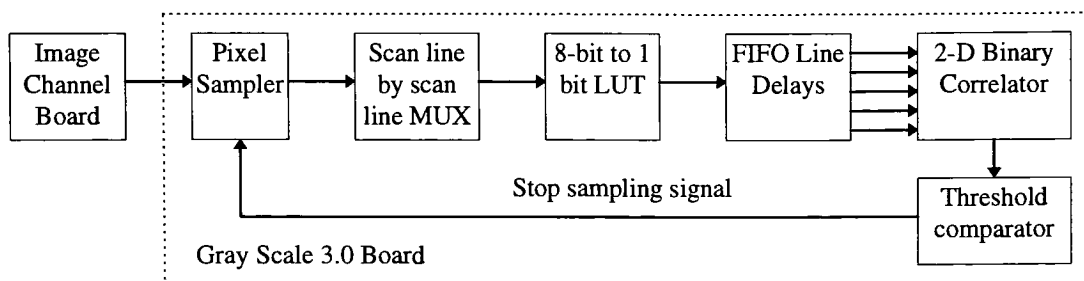
A desire originating from on site experimentation is to display unenhanced gray scale data and convolved gray scale data simultaneously. In Figure 2-29, the Convolver 3.0 allows selection of convolved/barrel shifted/clipped data from one of the four FIR data paths on the board, and passes this 14 or 16 bit data into a LUT. Data is transformed through this software configured LUT function into 8 bits per pixel. It is then provided on the address lines of the output VME bus in parallel with the combined phase data on the data lines. The Image Channel board captures the address line data, and provides it to the Gray Scale Display Board. On a different Image Channel board, the input data to the Convolver 3.0 is captured, and fed through a different data path on the Gray Scale Display Board. Data delays on the Convolver 3.0 force a four scan line delay between these two data streams, resulting in a four scan line shift on the video display. With the

understanding of this data shift, it is therefore possible to view the unenhanced and enhanced view of the same imperfection on the display.



**Figure 2-29: Capture of Enhanced and Unenhanced Data**

On the new board, software still provides the ability to stop the acquisition of new frames of data when a user sees a frame on the display they wish to keep. To assist in capturing a visual representation of what features look like, a “Feature Finder” function is provided. See Figure 2-30. Data received into the display board is passed through a software configured LUT, providing a single output (binary) bit for each pixel. This value is then run through a two-dimensional binary correlator, and the output compared to a software configured threshold limit. If the threshold is exceeded, a trigger is sent to the capture circuitry to stop capturing data. This allows the Display Board to run in a continuous capture mode, and automatically stop when appropriate parameters have been met. Software must configure the threshold and binary correlator to detect the appropriate features.



**Figure 2-30: Feature Finder Implementation**

### **3. Improvements**

#### **3.1. Logic Compaction**

Boards designed for use in the Veredus System are usually very dense. Over 90% of board real estate is typically employed on large designs. Additional functions or capabilities are always requested during the design cycle on a board, driving the density value higher. The software integration of a board, and subsequent use on site derives new needs for future systems and boards. This results in a continual need to place more functions in a smaller area.

Veredus designed boards utilize almost exclusively through hole technology devices. This conscious decision stems from the low volume of each board design that is manufactured. At most 50, and more typically 10 to 20, copies of each board design are required per year to meet system sales. Most surface mount assembly houses prefer to work with quantities much larger than this when stuffing designs, because of long set-up times for each different type of board run through the production line. Emphasis is therefore placed on manufacturing boards which have larger quantities by the assembly houses. This results in high per board manufacturing costs, and uncertain scheduling times for the fabrication of surface mount boards. Through hole designs allow more flexibility in the quantities of each type of board, while still providing reasonable delivery times and cost. By socketting almost all components as well, damaging a board during re-work is minimized through the use of through-hole technology. The use of the through hole technology comes at a price: employed components are typically larger than their



surface mount counterparts, resulting in larger boards and/or higher board densities. Despite the cost and scheduling issues, Veredus designed boards could benefit by pursuing surface mount technology in some areas.

Recently designed Veredus circuit boards construct glue logic mostly from programmable logic. This is partly done to minimize the number of circuit board modifications which are needed should a change in functionality be required. The largest reason is to minimize consumed board space. Discrete logic (i.e., AND, OR, and INVERT) devices are employed only to provide unique capabilities such as open collector outputs. Tri-state buffer and latch capabilities have been incorporated into programmable logic as well, allowing additional compaction of functionality into a smaller board space. Where possible, commercially available LSI and VLSI devices are employed on boards to shrink real estate requirements. The next logical step to reduce the physical size of implemented circuits is to pursue either ASIC implementations, or larger programmable logic devices.

In some situations, ASIC implementations allow the reduction of entire circuit boards into a single device. The massive use of static RAMs for various tables in the Veredus pipeline prevent this promise from becoming reality. Boards dealing with image data typically employ a static RAM for roughly one out of four devices in the pipelined data path. These RAM tables range in size from 4K x 8 to 256K x 4, requiring a minimum of 32K bits in each RAM which is implemented. Static RAM table implementations within gate arrays average six to seven equivalent gates for each stored bit [14], requiring 192K gates just to implement the smallest RAM table. Present technology allows in excess of

300K gates per gate array device [14], but one to two orders of magnitude increase would be required to implement the 5 to 15 tables which exist on large Veredus circuit boards. Even the implementation of a single RAM table requires a relatively large gate array, limiting options on where gate arrays could be implemented. Larger fabrication houses must be employed, with significant Non-Recurring Engineering (NRE) charges (over \$100K) and large minimum quantity purchases of devices (5,000 units). The use of semi-custom designs allows a higher internal RAM capability, but not large enough to implement more than one RAM table with controlling logic. Plus, semi-custom devices have higher NRE charges, and the same minimum quantities. As stated above, the Veredus marketplace needs at most 50 copies of a particular board per year. Pursuing an internal RAM gate array solution is only feasible if the implemented function can be employed multiple times on the same board, or can be employed on multiple boards. Pursuing a semi-custom design is even less likely.

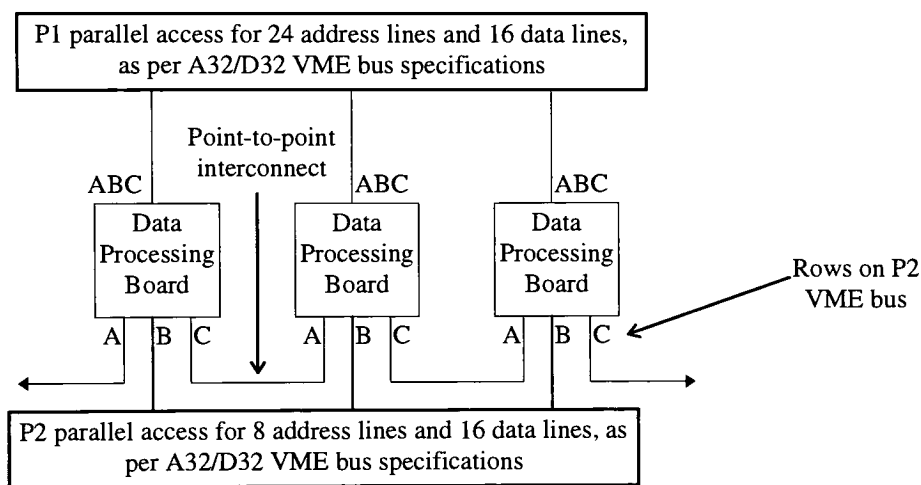
The simple compaction of logic functions offers a different story. Field Programmable Gate Arrays (FPGAs) readily exist, which can implement up to 5,000 equivalent gates. Nearly all logic functions other than the RAM tables and external bus (i.e., VME bus) signal buffers could be reduced into a series of these devices, significantly reducing required board real estate. Further, both small and large fabrication houses provide the transformation of the FPGA logic into gate array implementations for reasonable NRE charges, and relatively small minimum quantities. Implementation of functions that could be used on many board designs, therefore, require consideration for gate array implementation.

### **3.2. Backplane Communications**

The multiple P1 VME bus data transfer architecture served the purpose of the Veredus system initially conceived in 1984 well. Lessons learned from the first generation of boards required subsequent improvements that significantly increased the amount of data, which must be transferred from one function to another. With a rational desire to maintain a backward compatibility to the implemented P1 VME architecture, the amount of data that the VME busses can transfer mandated a significant increase in the number of functions on each board. This has resulted in large, dense boards, and a system interconnection scheme which only allows new sequential functions to be added by reworking existing board designs. Movement to a different interboard backplane communication scheme could allow smaller boards to be implemented, by allowing better division of functions between multiple boards. Easier integration of new board functions could also be accomplished with a new scheme.

In [7], a point-to-point interconnection scheme is used to connect the processing boards. A similar concept was employed on boards designed at Veredus Products Division. The project using these boards required functions that are above and beyond the general web scanning case. Both these implementations retained a VME bus for the system control. Instead of the classic Veredus System use of only P1 type VME busses, the new schemes employ the P1/P2 VME bus, as shown in Figure 3-1. User defined pins on the A and C rows of the P2 bus are defined with the point-to-point interconnections. Data from each processing board flows out the C row pins, through a custom designed backplane, and

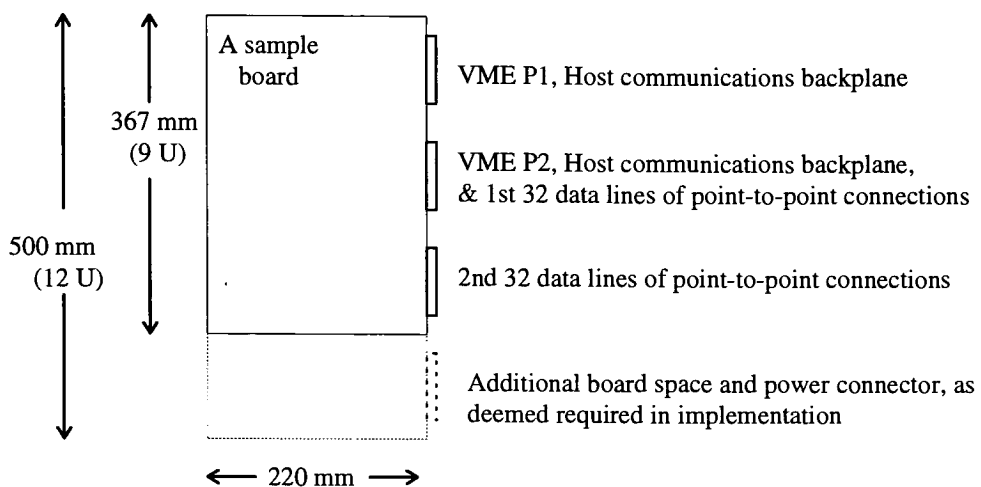
into the A row pins of the next board. Data connections stop at the A row of the second board, and the C row of the second board starts a new data connection. The data transmissions change from being an asynchronous bus as used on the VME P1, to a synchronous bus with one common clock across all data processing boards. On each cycle of the clock signal, the validity of data is indicated with a single bit, showing that data does, or does not exist, during the cycle.



**Figure 3-1: Conceptual Point-to-Point Interconnection**

An ideal implementation of this scheme maintains the common synchronous clock, but increases the number of communication signals. The data valid/no data signal is required, but the addition of data type flags allow boards to pass control information in the data stream. Examples of these signals are "last pixel of the line", and "event information" tag. To make provisions for faster circuits, the common clock should operate at 40 MHz, allowing for future enhancements to double the rate at which data can move through the system, compared to the present Veredus.

The transfer of information from one board must always proceed into the board following the present board, and stop. If the same data is to be transmitted to a board two slots away, the second board must retransmit the first board's data, as well as output its own. In addition, the P1 VME architecture demonstrated a board may need to emit multiple data streams, because of multiple enhancement functions on the board. For this reason, the data lines of the point-to-point connection need to be divided into multiple logical data busses. Each logical data bus needs all three control signals (data valid, last pixel, event tag), resulting in an implementation with three logical busses only receiving 22 signal lines between the three busses. Present Veredus system applications shows this will likely be insufficient. Hence, two successive vertical backplanes of the point-to-point interconnect should be allocated, as shown in Figure 3-2. This allows a significant increase in the amount of data which may be handed from board to board. Each board design becomes a minimal 9U implementation, with the top backplane connector the VME P1. Allowing provision for up to 60 bits of data from one board to the next at 40 MHz should handle the upgrade ability path for over 10 years.



**Figure 3-2: Sample Point-to-Point Interconnection Board**

In order to assure consistency of the point-to-point interface, implementation of a gate array for the interface to the point-to-point backplanes should be considered. All boards that interface to the point-to-point backplane will require the gate array, justifying the cost for the implementation. Through a standard Host processor accessible addressing scheme, this gate array should be configurable to variable width logical data busses on the point-to-point connections. Maximum system modularity is obtained in this manner, and minimum software integration issues as well, because all board to bus configuration is identical.

A point-to-point architecture creates a more modular system. Implementation of a board with a new function on it is possible without adding the new function to an already existing design. Boards that should have functions following the new one are simply moved over by one slot in the chassis, and the new board is slid into place. This provides a capability the current Veredus does not possess. The ability to modularly employ

additional image enhancement concepts becomes a reality. A board allowing single-pass erosion/dilation could be implemented using either a binary template matcher or a rank value chip from [11]. An IIR filter board could be developed with a more sophisticated filtering capability than the one in [7], and substituted for the classic FIR filter boards when conditions warrant. Different enhancement boards could feed into each other, i.e., an IIR performs a high pass for which it is suited, and these results are sent through an FIR.

In addition, the number of data lines allocated for board communications has increased significantly. This potentially allows functions of some existing boards to be implemented in a more atomic manner. Splitting the convolution and thresholding functions of the Convolver 3.0 is a possibility.

The feasibility of implementation is reasonable. The use of Euro-card standard sizes has been maintained, allowing a reuse of any previously installed system chassis. All backplanes in the image processing rack can be removed, and new point-to-point backplanes installed in the system. This can probably be performed on site in about four hours when needed. The redesign of the entire pipeline board set must be completed prior to starting field installations with any of the new circuitry. In actuality only small portions of the designs must be reworked if present board functions are sufficient to start with. The standard use of synchronous clocking and HSS/HSM circuits should provide a fairly clean transition to the new point-to-point concept gate array interface. Designs that require massive rework due to not having HSS/HSM concepts require rework to account

for other technology advances. To provide a complete backward compatibility path, a fairly trivial interface board would also be needed to allow transition from the new architecture to the old one.

### **3.3. Connectivity and Measurement**

A common bottleneck of nearly all web scanning systems is connectivity and measurement. The Veredus System is no exception. Improving the rate at which the Veredus can perform the connectivity and measurement process, and increasing the number of measurements that are performed, are significant areas for improvement. There are several methods to pursue for implementing the improvements.

The lowest implementation cost is to provide software to control the present Connectivity and Measurement Board. Since the hardware has been tested, and the rudimentary software written to perform the measurements, the implementation is strictly a software integration issue. Appropriate cross-web offset parameters must be provided to the C&M processor to account for web weave, Host level drivers must be implemented, and interprocessor communications protocols worked out. The Quad-II board could be reprogrammed for a different use, or removed from the system. It is estimated this would provide roughly a tenfold increase in the number of features, which could be handled per lot time compared to the Quad-II implementation. Because the board only handles seven non-background phases, and the use of the combiner table on the Convolver 3.0 could provide up to 255, this implementation would simply be faster, and provide only a little more information about each feature.



Reworking the C&M board to account for the 255 possible values from the Convolver 3.0 and RLE 1.6 would provide a significant increase to the information gathered on each feature. A slight rework of the RLE 1.6 board would also be required to expand from four bits of phase information in the RLE-II encoding scheme to eight. This allows accurate reporting of the combiner 8-bit output values to the reworked C&M board. With density and speed increases in programmable logic, many of the register locations implemented in octal devices on the C&M board for speed could be combined into a few large devices. This should offer a dramatic decrease in space, affording fairly easy addition of the additional phase values. After analysis with compacting other functions on the board, it may be possible to add a second processor on the C&M. This second processor would receive the measured outputs of the first, and perform additional classification. This solution employs the functional advances that have been implemented on boards further up the pipeline. The solution also potentially offers a higher system through-put rate than the presently designed C&M board by allowing extra classification. There is still a limitation, however, that data must be run length encoded prior to connectivity and measurement. Plus, the rate of the connectivity and measurement process is still two orders of magnitude slower than the pixel rate.

In [15], an APA512+ board is described, which can perform connectivity and measurement on data at up to 25 MHz, with a more typical rate of 10 to 20 MHz. It could be possible to employ this board for performing the connectivity and measurement process in the Veredus system. Continuous flow-through data must be transformed into

frames for use on this board. Resulting data must then be extracted from localized memory locations on the board, and offset values adjusted for web weave and the edge tracking process. To account for the real-time requirements of the Veredus, means would have to be provided to divide the existing 20 MHz data rate (or 40 MHz in an improved system) into multiple independent 10 MHz data streams to assure the APA512+ can handle the continuous data flow. Analysis is required to understand if the reported 30,000 feature measurements per second on each board is sufficient for the Veredus application. Verification the APA512+ handles multiple phases in the connectivity process is required. The advantage of this solution is the elimination of run length encoding; the disadvantage is the external formatting which is required to employ the external design. In reality, only the acquisition of the feature extraction chips, and placement on a continuous flow-through architecture board, would be practical for use in the Veredus.

Modular cellular logic processing elements are identified in [8], which allow implementation of many different image processing functions. Of most interest is the suggested implementation of the connectivity process. These circuits are proposed to operate with data processed at up to 10 million pixels per second. The described implementation employs an iterative loop to perform connectivity from binary, but not run length encoded, image data. Because it is iterative, and the number of loops is dependent upon the feature being connected, the implementation does not meet the real-time requirements for the Veredus. If an upper bound could be established on the number of iterations required for features found in the Veredus, a real-time constraint could be

established. The aggregate system data rate could then be vertically striped as needed to reduce the individual circuit data rate to what the processing element can handle. For instance, if each processing element can handle data at 2.5 MHz, and the aggregate data rate is still 20 MHz, eight vertical stripes would be required. Because each processing element is a single 40 pin device, placing eight of these devices on a single board is feasible. Detailed analysis is required to establish the workings of the iterative algorithm, given the features that Veredus detects and reports.

### **3.4. Classification**

The Veredus System presently performs classification in software. Configuration tables can be loaded on pipeline boards to allow combination of multiple enhancement paths, which allows a prioritization of enhancement methods. This allows some form of classification on the pipeline boards. However, these Convolver 3.0 tables are still configured by software, resulting in effectively a software classification. Methods to accelerate classification methods would improve the Veredus System.

Neural networks seem a logical solution to the classification problem. Feature patterns seem to vary slightly from a nominal feature, and neural networks are good at interpreting small variations, and reporting high probabilities of a match. Length, width, and area of each phase of a feature are calculated in the Veredus at present, which would serve as a good baseline for the neural network input. Adding fields such as perimeters, and reporting the measurements on the same feature from multiple enhancement paths would allow additional confidence in the network with increasing number of input nodes. A significant advantage of the neural network could be the ability to analyze multiple features simultaneously. This allows features that tend to be broken up by the enhancement process to be analyzed and reported as one common classified feature, instead of a bunch of small localized dots.

Implementation of the neural network can occur in either software or hardware. The software situation would simply require an additional processor within the image processing pipeline. Use of a second processor on the Connectivity and Measurement

Board is a possibility. A hardware accelerator could also be constructed allowing a series of pipelined multipliers, accumulators, and RAM tables. Prior to commitment to an employed scheme, realistic input parameters for the neural network need to be established, and simulated for processing time in the software solution. The pursuit of a hardware specific application only seems reasonable if the software processing time for each feature is too long. The experiences of [16] demonstrate a hardware accelerator will eventually be mandated.

Training of the neural network will be similar to the present system training. A series of imperfections must be gathered, viewed by the human inspectors, and classified. Then, instead of feeding the inputs to a rule based classification tool, the inputs would be employed in training of the network.

An additional means of assisting with classification is to provide the rule based algorithm on a lower level slave processor. The classification task could be distributed to allow classification of all information from given pipeline to occur within that pipeline. Then the higher level Host needs only to resolve interpipe classification. The pipeline classification could be performed by a second processor on the C&M board, or a Quad-II board, which is no longer employed for the connectivity and measurement process. This would require additional complexity in system configuration to establish which portion of the classification should be performed on the Host, and which portion on the slave processor. It would seem plausible most of the configuration could be hidden from the end user.

### **3.5. Diagnostics**

Functional digital electronics diagnostics on Veredus Products Division Boards have developed to a reasonable level. These capabilities allow processor and time intensive algorithms to verify proper operation of the boards by disabling the scanning system operations and running an off-line test. This is excellent for detailing all operations of the boards, and assisting in identification of which particular part of a board is faulty on a test bench. However, production line operators have indicated a need for a fast health check capability in a web scanning system.

Experimentation with Built-In Self Test (BIST) was started on the Convolver 3.0. Continuation of this concept should be expanded to all boards employed in the Veredus. BIST pattern generators should pass data on the board-to-board interconnection busses, allowing slave boards to receive the results through a Signature Analyzer and assure proper bus operation. Internal BIST operations should have two modes: a destructive test, and a non-destructive test. In the destructive test, the BIST data generator is allowed to write new data into RAM tables, local registers, etc., to fully verify the operation of the board. Once the RAMs and registers are loaded, the LFSR patterns are switched to run through the configured tables, resulting in a pattern of data coming from the RAMs which should be a known value. Signature Analyzers can then verify the operation of nearly all board functions. The price of this test is the destruction of any configurations that exist in the RAMs or registers. System software must then reload the proper values into the tables, which will require some amount of time. A non-destructive test does not touch the

contents of RAM tables or configuration registers, resulting in no reconfiguration once the test is complete. A full verification of all the RAMs is not possible this way, but nearly all data processing paths can be checked within about two seconds in the non-destructive mode. System software could automatically run the non-destructive test at the end of each product roll, providing some level of constant assurance the system is operating properly. A full destructive test could then take a few minutes once a shift to verify proper operation of the RAM tables as well, with all but a few seconds allocated to reconfiguring the pipeline for operation.

A twist on the BIST capability is to verify the system response to an optical signal. Veredus System optics are tested by placing targets on the inspection roller at routine shut-down intervals. These typically occur once every two weeks, or even less frequently. Again, production line quality assurance personnel would like a verification once a shift, or once a day. Attempts at working with customers to generate mechanical devices that can be automatically inserted in front the camera have not been successful. Optical verification needs the ability to disable the normal illumination source by the Host computer (via Digital I/O), and enable an alternate illumination source. This second source needs a well controlled illumination level, to verify the camera lens and CCD detector are working properly. It may be necessary to reduce the line transfer pulse rate during this test, or in the case of constant pixel mode, switch to a constant frequency mode, to assure sufficient signal can be obtained. Sample imperfections must somehow be illuminated onto the inspection plane during this process, with one to two pixel accuracy, and illumination variations accurate to less than 1%. Recent advances in laser

diode capabilities show promise of allowing this equivalent function to be implementable. Appropriate sensor signals must then be received through the system, and verified to be within a few percentage points of the expected pattern.

### **3.6. Display Subsystem**

The present Veredus system has implemented a feature finder option on the Gray Scale Display Board. The implementation thresholds data to a binary level, and then employs binary template matching to determine if a feature exists. A more accurate means to accomplish this function is to provide a feedback path from the classification output. The entire processing power of the Veredus detection system can be employed to establish if an desired imperfection exists. This eliminates the completion of a duplicate software configuration for the Gray Scale Display Board, and assures consistent system imperfection detectability.

To implement this function, the connectivity, measurement, and classification process must meet a real-time constraint. Once the feature enters the gray scale display board, it will be written into display memory. The memory write pointer will then increment to pixels and scan lines following the feature. Eventually, the write pointer will loop around and need to overwrite the memory location that contains the desired feature. By the time this occurs, the classification process must be complete to stop the acquisition of data, and maintain the feature. With a 1024 line display memory used on the Gray Scale Display 3.0 Board, and a minimal realistic line length of 512, a 20 MHz data rate allows 26 milliseconds to perform the frame stop. To make the constraint more stringent,



writing to the display should cease when the feature is no more than three quarters of the way through the display memory, or at around 20 milliseconds.

Exact timing requirements for the Veredus System to accurately classify an imperfection cannot be discussed due to confidentiality. It can be stated the response time is larger than 20 milliseconds. Implementation of this improvement is dependent upon success in implementing a connectivity, measurement, and classification improvement. If these other improvements reduce the system latency sufficiently, this change is as simple as removing the feature finder option on the Gray Scale Display 3.0 Board, and adding the remote stop acquisition cable. If it is not reduced sufficiently, the memory space on the Gray Scale Display Board could be increased as needed to reach the appropriate system real-time response.

### **3.7. Triple Banked Memories**

Double banked, or ping-pong, RAMs are used in many locations in the Veredus system. In many situations, controlling software has to keep a copy of the board RAM contents in its local memory to generate new values. For instance, DCM gain values are held in the processor's local memory so that an IIR low pass function can be implemented in the calculation of new coefficient values. The Host has to maintain copies of all 16 logical threshold banks for each physical threshold bank on the Convolver 3.0, because any one changing is not supposed to affect any others.

This becomes burdensome for two reasons. First, additional memory at the processor is consumed by the duplicate bank. Second, additional processor cycles are consumed copying the established table from the processor's local memory to the image processing board's LUT. Implementing what will be called a triple banked memory can resolve these issues.

Each LUT, instead of having two RAM banks, has three. In the scanning mode, system software only has access to one bank. This table is filled with the appropriate LUT function required, and software sets a bit on the image processing board indicating to make the table active. Rather than the board placing this RAM on-line, a state machine takes control of the RAM. One of the other two RAM devices is off-line, and the other is on-line in the image processing path. The state machine copies the contents of the software accessible table into the off-line bank. It then switches the off-line and on-line bank at the start of the next scan line. Meanwhile, the software accessible RAM is still

available for microprocessor access, with the same values that were written prior to the activation bit being set.

This improvement seems to add significant complexity to board designs. Yet logic required to implement state machines becomes more compact in programmable logic all the time. Memory densities used on the image processing boards also continue to increase. So while microprocessor speeds continue to become faster, they must still spend the same percentage of their time, if not more, double writing these RAM tables. Plus, increasing amounts of processor memory is consumed with the local copies of each table. This can be overlooked when the tables are small, but if four parallel copies of 256K threshold tables must be stored for each of four boards, this translates to four megabytes of memory on the Host processor. In a real-time web scanning system, where all processor cycles are needed, and all memory should be accounted for, this is an improvement software engineers request.

## 4. Conclusion

The Veredus Quality Control System is successful at scanning web products. The production line experiences which have been incorporated in the system hardware design have resulted in a robust industrial grade image analysis engine. Equipment installed on the production line has been minimized to allow negligible impact on physical space requirements. Image enhancement and imperfection detection can be carried out a significant distance from the production line, in a customer's computer room. Image data is processed in a continual flow-through manner, meeting the requirements of a real-time system.

The Veredus Quality Control System is intended for the most demanding web scanning applications. This has resulted in a higher complexity and cost than most commercially available image processing systems. It has also allowed successful application of the system to web scanning situations where most commercially available systems fail.

The largest challenges facing the Veredus Quality Control System are finding ways to process data faster, with better detectability. Present bottlenecks in the system impeding these items are the interconnecting backplanes and the connectivity, measurement, and classification process. The proposed changes provide a way to eliminate the backplane bottleneck for reasonable development costs, while maintaining some level of backward compatibility. Of the discussed connectivity, measurement, and classification concepts, a rework to the Connectivity and Measurement Board is most feasible from a development

standpoint. The addition of a second processor on the board allows a test bed for experimenting with neural networks, and boosts the system feature load capability for a relatively small development effort. A new development project could then be considered to refine the other connectivity concepts for the real-time requirements of a web scanning system.

## 5. Literature Cited

- [1] Robison, Stanley L. and Miller, Richard K., *Automated Inspection and Quality Assurance*, Quality and Reliability Series (New York: Marcel Dekker, Inc.), 1989.
- [2] J. W. Roberts, S. D. Rose, G. Jullien, L. Nichols, P. T. Jenkins, S. G. Chamberlain, G. Maroscher, R. Mantha, and D. J. Litwiller, "A PC-based Real Time Defect Imaging System for High Speed Web Inspection," in *Machine Vision Applications in Industrial Inspection*, Fredrick Y. Wu, Benjamin M. Dawson, Editors, Proc. SPIE 1907, pp 164-176 (1993).
- [3] Laplante, Phillip A., *Issues in Real-Time Image Processing*, Proceedings of 1993, International Conference on Systems, Man, and Cybernetics, volume 2 (New York: IEEE), 1993.
- [4] Macaire, L. and Postaire, J.G., *Real-Time Control of Galvanized Coating Aspect by a Texture Inspection System*, Proceedings of 1993, International Conference on Systems, Man, and Cybernetics, volume 2 (New York: IEEE), 1993.
- [5] Garcia, Daniel F., del Rio, Marcos A., Diaz, Jose L., and Suarez, Francisco J., *Flatness Defect Measurement System for Steel Industry Based on a Real-Time Linear-Image Processor*, Proceedings of 1993, International Conference on Systems, Man, and Cybernetics, volume 3 (New York: IEEE), 1993.
- [6] Dragana Brzakovic, and Hamed Sari-Sarraf, "Automated Inspection of Nonwoven Web Materials: A Case Study," in *Machine Vision Applications in Industrial Inspection II*, Benjamin M. Dawson, Stephen S. Wilson, Fredrick Y. Wu, Editors, Proc. SPIE 2183, pp 214-223 (1994).
- [7] Joseph D. Burjoski, "Novel Hardware Architecture for Real-Time, Continuous Line Scan Processing," in *Machine Vision Applications, Architectures, and Systems Integration III*, Bruce G. Batchelor, Susan Snell Solomon, Federick M. Waltz, Editors, Proc. SPIE 2347, pp 340-351 (1994).
- [8] Jonker, Pieter P., Komen, Erwin R., and Kraaijveld, Martin A., *A Scalable, Real-Time Image Processing Pipeline*, Machine Vision and Applications, volume 8, number 2, pp 110-121, 1995.
- [9] Dennis C. Mills, "A Modular System for Automated Surface Inspection" in *Machine Vision Applications in Industrial Inspection*, Fredrick Y. Wu, Benjamin M. Dawson, Editors, Proc. SPIE 1907, pp 13-19 (1993).
- [10] *1994 Dalsa data book*, Dalsa, Inc. Waterloo, Ontario, Canada, August 1993.

[11] *LSI Logic Digital Signal Processing (DSP) Databook*, Milpitas, California, June 1990.

[12] Pink, Jeffery R., *Features and Neural Net Recognition Strategies for Hand Printed Digits*, RIT Computer Engineering Thesis, October 30, 1995.

[13] Joon H. Han, Doo M. Yoon, and Myeong K. Kang, "Features for Automatic Surface Inspection" in *Machine Vision Applications in Industrial Inspection*, Fredrick Y. Wu, Benjamin M. Dawson, Editors, Proc. SPIE 1907, pp 114-123 (1993).

[14] *Amtel Corporation CMOS Gate Array Design Manual*, San Jose, California, September, 1993.

[15] *APA512+ Feature Extraction Processor*, Atlantek Microsystems, Adelaide, S.A. Australia.

[16] C. Sanby and L. Norton-Wayne, "Machine Vision Inspection of Lace Using a Neural Network" in *Machine Vision Applications in Industrial Inspection III*, Fredrick Y. Wu, Stephen S. Wilson, Editors, Proc. SPIE 2423, pp 314-322 (1995).

[17] Oppenheim, Alan V. and Schafer, Ronald W., *Digital Signal Processing*, (Englewood Cliffs, New Jersey: Prentice Hall), 1975.

[18] Giardina, Charles R. and Dougherty, Edward R., *Morphological Methods in Image and Signal Processing*, (Englewood Cliffs, New Jersey: Prentice Hall), 1988.

[19] Haykin, Simon, *Neural Networks, A Comprehensive Foundation*, (Englewood Cliffs, New Jersey: Macmillan Publishing Company), 1994.